

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Service, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY) 31-03-2000		2. REPORT DATE Final report		3. DATES COVERED (From - To) 03-1997 - 03-2000	
4. TITLE AND SUBTITLE Hp-meshless cloud method for dynamic fracture in fluid structure interaction				5a. CONTRACT NUMBER N00014-95-C-0373 (-96-C-0329)	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER SBIR N95-089	
				5d. PROJECT NUMBER	
6. AUTHOR(S) Tadeusz J. Liszka (PI) C.A.M. Duarte O.P.Hamzeh				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Altair Engineering (formerly Computational Mechanics Co) 7800 Shoal Creek Blvd, Suite 290E Austin, TX 78757				8. PERFORMING ORGANIZATION REPORT NUMBER TR-99-05	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Office of Naval Research 800 North Quincy Street Arlington Virginia 22217-5660 Roshdy S. Barsoum, ONR 334				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSORING/MONITORING AGENCY REPORT NUMBER	
12. DISTRIBUTION AVAILABILITY STATEMENT Approved for public release, SBIR report, distribution unlimited					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT The project aimed at the development of a new meshless technology, and computer code based on, for the analysis of crack propagation in dynamics fluid structure interaction. In the meshless formulation crack propagation is not restricted by the existence of domain discretisation (meshless method) and suitable error estimators combined with adaptivity assure highly accurate solution. A general purpose, three dimensional meshless analysis code, called PHLEXcrack has been developed, tested and delivered to Navy. At present the code is capable of analysis of static and dynamic structure responses including the underwater shock analysis (provided via interface to "U.S.A." code) with dynamic crack propagation through the selected part of the domain. Crack propagation criteria for brittle materials is provided based on least square fit method and energy release rate data. Users can specify own routines for alternate fracture criteria and crack propagation prediction. Full suite of classical finite elements (comparable to basic Nastran capabilities), together with hp-adaptive volume finite elements, and 3 dimensional meshless formulation are implemented and can be freely mixed to obtain efficiently the solution of displacements, stresses and fracture measures in the structure. Nonlinear capabilities of the code consist of large deformations, nonlinear materials, (incl. plasticity), and contact with rigid surfaces.					
15. SUBJECT TERMS dynamic fracture, crack propagation, meshless methods, adaptivity, fluid-structure interaction					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			W. W. Tworzydło
none	none	none	none	8 + 98	19b. TELEPHONE NUMBER (Include area code) (512)-467-0618

200428137

March 2000

Hp-Meshless Cloud Method for Dynamic Fracture in Fluid Structure Interaction

- Final Report -

Prepared By : The Computational Mechanics Company, Inc.
Prepared For: Office of Naval Research
Contract No.: N00014-95-C-0373

DTIC QUALITY INSPECTED 3

20000428 137

Contents

1	Introduction	1
2	Formulation of Element Partition Approximations	2
2.1	Finite Element Partition of Unity	2
2.2	Shepard Partition of unity	3
2.3	Element Partition Shape Functions: The Family \mathcal{F}_N^p	5
2.4	Construction of Discontinuous Element Partition Shape functions	7
3	Customized Shape Functions for an Edge in 3D	11
4	A Posteriori Error Estimation for The EPM	14
5	Extraction of Stress Intensity Factors: The Least Squares Fit Method	15
6	Crack Evolution Models	18
7	Dynamic Flow-Structure Interaction Problems	20
7.1	Introduction	20
7.2	A General Problem Statement	23
7.3	Structural Deformation Problem	24
7.4	Flow Equations; an ALE Description	28
7.5	Flow-Structure Interaction Equations	30
8	Solution of the Flow-Structure Interaction Problem	32
8.1	Time Integration for the Structural Domain	33
8.2	Time Integration for the Flow Domain	34
8.3	Solution of Combined Flow-Structural Equations	34
8.4	Moving Mesh Algorithm	35

8.5	Overall Solution Algorithm	35
9	Underwater Shock Analysis	36
9.1	Introduction	36
9.2	Underwater Shock Analysis Code	37
9.3	Structural Response Equation	38
9.4	USA-PHLEXcrack Interface	39
9.5	Computational Procedure	40
9.6	Software Validation	41
10	Mixed Elements in PHLEXcrack: EPM and Classical Shell Elements	41
11	User Routines for Crack Propagation	41
12	Controlling the accuracy of numerical algorithms	42
12.1	Crack details and element sizes	42
12.2	Crack shape and accuracy of SIF estimation	42
12.3	Crack smoothness	42
12.4	Crack folds	43
13	Numerical Examples	43
13.1	Single Crack with Mode I Solution Under Static Loading	44
13.2	An Inclined Crack Problem	52
13.3	Plate Under Impact Load	61
13.3.1	Reference Solution	62
13.3.2	Stationary Crack	66
13.3.3	Moving Crack with Prescribed Speed and Direction	66
13.3.4	Moving Crack with Prescribed Speed	68
13.4	Edge Cracked Plate Under Asymmetric Impact	75

13.5 Welded T-Joint with a Crack	76
13.6 Welded T-Joint with a Fully Embedded Crack	80
13.7 Section of Infinite Cylinder	87

List of Figures

1	Global finite element shape function N_α built on a mesh of triangles and quadrilaterals.	3
2	Overlapping patches corresponding to clouds ω_α and ω_β . Polynomials of differing degree p_α and p_β can be associated with nodes at \mathbf{x}_α and \mathbf{x}_β so as to produce non-uniform hp -cloud/finite element approximations.	4
3	Example of element partition discretization cut by a crack	8
4	Coordinate systems associated with an edge in 3D space.	11
5	Flowchart of PHLEXcrack TM	16
6	Domain used for the Least Square Fit method.	17
7	Local unit vectors at crack vertex.	19
8	A general statement of the flow-structure interaction problem.	24
9	Transient structural deformation at large deformations.	25
10	Fluid-structure interaction: complementary subdomains and interface conditions.	31
11	Single edge-noth test specimen	44
12	CEPM discretization for the model problem shown on Fig. 11.	46
13	Triangularization of the crack surface.	47
14	Zoom at the crack front.	47
15	Displacement in the y -direction computed with discretization WA-3.	49
16	Von Mises stress computed with discretization WA-3.	50
17	Zoom at the crack front showing von Mises stress computed with discretization WA-3.	50
18	Zoom at the crack front showing von Mises stress computed with discretization vis-5.	51
19	Problem definition.	53
20	CEPM discretization for the model problem shown on Fig. 19.	54
21	Zoom showing the elements cut by the crack surface.	55
22	Three-dimensional view of the crack front.	55

23	Contour plot of the displacement in the vertical direction computed using the discretization WA-3.	56
24	Contour plot for the von Vises stress field computed with discretization WA-3.	57
25	Zoom showing the von Mises stress near the crack front computed with discretization WA-3.	58
26	Zoom showing the von Mises stress near the crack front computed with discretization vis-5.	58
27	Model problem used to analyze the performance of the CEPM in modeling propagating cracks in a body subjected to impact loads.	63
28	CEPM Discretization.	63
29	Crack Representation.	64
30	CEPM Discretization.	64
31	Definition of reference problem.	66
32	Time history for mode-I and mode-II stress intensity factors K_I and K_{II} using Discretizations 1 and 2. Stationary crack.	67
33	Time history for the energy release rate G using Discretizations 1 and 2. Stationary Crack.	67
34	Time history for mode-I and mode-II stress intensity factors K_I and K_{II} using Discretization 1. Moving Crack with prescribed speed and direction.	68
35	Time history for the energy release rate G using Discretization 1. Moving Crack with prescribed speed and direction.	69
36	Time history for K_I and K_{II} using Discretization 1. Moving crack with advancement direction given by (10).	69
37	Time history for G using Discretization 1. Moving crack with advancement direction given by (10).	70
38	Crack surface at $t = 0.00168s$. Moving crack with advancement direction given by (10).	71
39	Time history for K_I and K_{II} using Discretization 3. Moving crack with advancement direction given by (10).	72
40	Time history for G using Discretization 3. Moving crack with advancement direction given by (10).	72

41	Comparison of results for K_I and K_{II} computed with Discretizations 1 and 3.	73
42	Comparison of results for G computed with Discretizations 1 and 3.	73
43	Time history for K_I and K_{II} using coarse mesh and $\mathbf{p}_c = \mathbf{0} + \mathbf{1}$, $\mathbf{p}_c = \mathbf{0} + (2, 2, 1)$ (Discretizations 3 and 4).	74
44	Time history for G using coarse mesh and $\mathbf{p}_c = \mathbf{0} + \mathbf{1}$, $\mathbf{p}_c = \mathbf{0} + (2, 2, 1)$ (Discretizations 3 and 4).	74
45	Time history for G using coarse mesh and $\mathbf{p}_c = \mathbf{0} + \mathbf{1}$, $\mathbf{p}_c = \mathbf{0} + (2, 2, 1)$ (Discretizations 3 and 4). The tic marks of the x-axis are placed exactly when the crack front crossed the boundary between two elements.	75
46	The grids used: (a,b) for EPM, and (c,d) for FE analyses.	77
47	Comparisons of Stress Intensity Factors obtained from EPM, FE, and analytical solutions.	78
48	Welded T-section beam with a crack.	79
49	EPM discretization for welded T-section beam.	81
50	Crack surface of welded T-section example at time 0.0015.	82
51	Crack surface of welded T-section example at time 0.0020.	82
52	Crack surface of welded T-section example at time 0.0030.	83
53	Welded T-section beam with a penny-shaped crack.	84
54	Top view of the penny shape crack in a welded T-section beam. (Shock waves are coming along Z-axis)	85
55	Penny shape crack in a welded T-section beam at times 0.0014, 0.0016, 0.0018, 0.0019	86
56	Domain and boundary conditions for the infinite cylinder problem.	87
57	Infinite cylinder model using classical finite elements and EPM ICells.	89
58	EPM subsection of mixed infinite cylinder model.	90
59	Infinite cylinder with the initial crack.	90
60	Infinite cylinder: Stresses in the model without the crack, time = $9e-5$	91
61	Infinite cylinder: X Displacements in the model with the crack, time = 0.0001	91
62	Infinite cylinder: Stresses in the model with the crack, time = 0.0001	92

63	Infinite cylinder: Crack growth at the time $t = 0.072$	92
64	Infinite cylinder: Crack growth at the time $t = 0.102$	93
65	Infinite cylinder: Crack growth at the time $t = 0.138$	93

1 Introduction

The meshless method implemented in PHLEXcrackTM is the so-called Element Partition Method (EPM) which is a variation of *hp*-Cloud [15–17, 30, 38] and Finite Element Partition of Unity [2, 3, 33] Methods. The key feature of the Element Partition Method is the use of a partition of unity to build the approximation spaces. This partition of unity framework has several powerful properties such as the ability to produce seamless *hp* approximations with nonuniform *h* and *p*, the ability to develop customized approximations for specific applications like dynamic crack propagation, the capability to build *p*-orthotropic approximations on, e.g., three-dimensional tetrahedral meshes, etc. Several of the so-called meshless methods proposed in the last years also make use explicitly or implicitly of a partition of unity to build the approximation spaces [4, 17, 31, 35]. The fundamental difference between these methods and the EPM is in the choice of the partition of unity. In the EPM the partition of unity is provided by a combination of Shepard [25, 28, 45] and finite element partitions of unity. This partition of unity allows the inclusion of arbitrary crack geometry in a model without any modification of the initial discretization. We call this partition of unity Cracked Element Partition of Unity (CEPOU). This choice of partition of unity also avoids the problem of integration associated with the use of moving least square or conventional Shepard partitions of unity which are used in several meshless methods [4, 9, 27, 29, 31, 35]. Performance studies show that the EPM is orders of magnitude more computationally efficient than other meshless methods. In fact, the same studies also show that the EPM is often more computationally efficient than the Finite Element Method, while being flexible enough to allow the modeling of dynamic crack propagation in complex three dimensional structures without any remeshing. In addition to that, the partition of unity used in the EPM allows ease implementation of essential boundary conditions which is a problem when, e.g., moving least square partitions of unity are used as in the Element Free Galerkin and Reproducing Kernel Methods [4, 31, 35].

An important feature of the EPM is the ease in which customized shape functions can be added to the set of interpolating functions. This feature can be very advantageous in cases where information about the solution and how it looks is known a priori, as in the case of re-entrant corners. For cracks, asymptotic singular functions are used to enrich the basis functions of the meshless nodes close to the crack tip with the order of singularity derived from analytical solutions [13, 14]. Using customized shape functions reduces the number of degrees of freedom needed to accurately represent the solution thus reducing time and storage requirements.

Once a solution is obtained at some step, the amount and direction of crack propagation over the next time increment can be estimated. These values are evaluated at a series of representative points along the crack front. The front is then advanced to its new position. The model is updated to accommodate the advanced crack. However, unlike with finite elements the crack can propagate freely in any direction and not necessarily along element faces. Moreover, if the crack cuts an integration cell it is not required to regenerate any new

cells or connectivities. However, for an accurate representation the integration rule will be enriched/modified within those cells that are cut by the crack. In addition, enrichment with the auxiliary singular functions will be maintained for the meshless nodes closest to the crack front.

The crack propagation problem usually requires complex geometric and topological descriptions and calculations. This serves several essential tasks such as: defining the crack surface and front, updating them, interaction of crack with outside boundary, discontinuities and integration around crack, etc. In order to obtain accurate and efficient geometric calculations, the geometric kernel of HyperMeshTM (a powerful geometry engine and pre/post-processor) was modified for our purposes, and appropriate interface functions were designed and coded. This kernel is being linked as a library into PHLEXcrackTM. PHLEXcrack code is based on a proprietary FEM environment **ProPHLEX**, which was during this project, expanded to handle the EPM and CEP methods [19].

2 Formulation of Element Partition Approximations

Let Ω be an open domain in \mathbb{R}^n , $n = 1, 2, 3$ and \mathcal{T}_N an open covering of Ω consisting of N supports ω_α (often called *clouds*) with centers at \mathbf{x}_α , $\alpha = 1, \dots, N$, i.e.,

$$\mathcal{T}_N = \{\omega_\alpha\}_{\alpha=1}^N \quad \bar{\Omega} \subset \bigcup_{\alpha=1}^N \omega_\alpha$$

The basic building block of an element partition of unity (EPOU) approximation is a set of functions φ_α defined on the supports ω_α , $\alpha = 1, \dots, N$, and having the following property

$$\begin{aligned} \varphi_\alpha &\in C_0^s(\omega_\alpha), \quad s \geq 0, \quad 1 \leq \alpha \leq N \\ \sum_{\alpha} \varphi_\alpha(x) &= 1 \quad \forall x \in \Omega \end{aligned}$$

The last property implies that the functions φ_α , $\alpha = 1, \dots, N$, are non-zero only over the supports ω_α , $\alpha = 1, \dots, N$. The functions φ_α are called a *partition of unity subordinate to the open covering* \mathcal{T}_N . Examples of partitions of unity are Lagrangian finite elements, moving least squares and Shepard functions [17, 25].

2.1 Finite Element Partition of Unity

In the case of finite element partitions of unity, the supports (clouds) ω_α are simply the union of the finite elements sharing a vertex node \mathbf{x}_α (see, for example, [33, 38] and Figs. 1 and 2)

and the partition of unity function φ_α is equal to the global finite element shape function N_α associated with a vertex node \mathbf{x}_α as depicted in Fig. 1. In this case, the implementation of the method is essentially the same as in standard finite element codes, the main difference being the definition of the shape functions as explained in Section 2.3. This choice of partition of unity avoids the problem of integration associated with the use of moving least square or Shepard partitions of unity used in several meshless methods. Here, the integrations are performed with the aid of the so called master elements, as in classical finite elements. Therefore, the EPM can use existing infrastructure and algorithms for the classical finite element method.

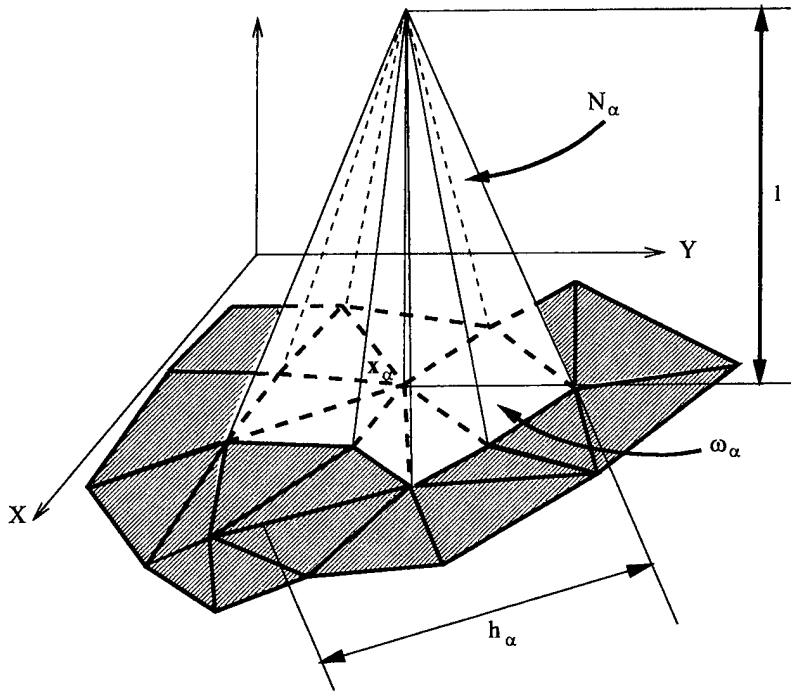


Figure 1: Global finite element shape function N_α built on a mesh of triangles and quadrilaterals.

2.2 Shepard Partition of unity

In this section we describe the construction of a partition of unity using the so called Shepard formula.

Let $\mathcal{W}_\alpha : \mathbb{R}^n \rightarrow \mathbb{R}$ denote a *weighting function* with compact support ω_α that belongs to the space $C_0^s(\omega_\alpha)$, $s \geq 0$ and suppose that

$$\mathcal{W}_\alpha(\mathbf{x}) \geq 0 \quad \forall \mathbf{x} \in \Omega$$

Suppose that such a weighting function is built at every cloud ω_α , $\alpha = 1, \dots, N$. Then

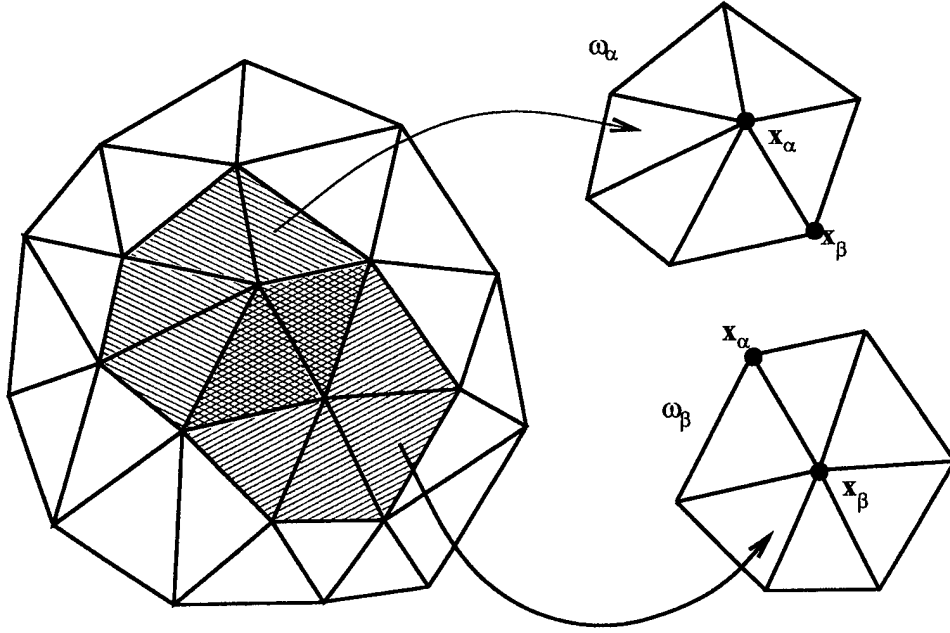


Figure 2: Overlapping patches corresponding to clouds ω_α and ω_β . Polynomials of differing degree p_α and p_β can be associated with nodes at \mathbf{x}_α and \mathbf{x}_β so as to produce non-uniform hp -cloud/finite element approximations.

the partition of unity functions φ_α associated with the cloud ω_α can be defined by

$$\varphi_\alpha(\mathbf{x}) = \frac{\mathcal{W}_\alpha(\mathbf{x})}{\sum_\beta \mathcal{W}_\beta(\mathbf{x})} \quad \beta \in \{\gamma : \mathcal{W}_\gamma(\mathbf{x}) \neq 0\} \quad (1)$$

which are known as Shepard functions [45].

Consider now the case in which the weighting function \mathcal{W}_α is taken as the global finite element shape functions N_α associated with the node \mathbf{x}_α , $\alpha = 1, \dots, N$. Let τ be a finite element with nodes \mathbf{x}_β , $\beta \in \mathcal{I}_\tau$ where \mathcal{I}_τ is an index set and let $\mathbf{x} \in \tau$. The only non-zero partition of unity functions at \mathbf{x} are given by

$$\varphi_\alpha(\mathbf{x}) = \frac{N_\alpha(\mathbf{x})}{\sum_{\beta \in \mathcal{I}_\tau} N_\beta(\mathbf{x})} = \frac{N_\alpha(\mathbf{x})}{1} = N_\alpha(\mathbf{x}) \quad \alpha \in \mathcal{I}_\tau$$

since the finite element shape functions form a partition of unity. Therefore, it is not necessary to use the Shepard formula to build the partition of unity when the weighting functions are taken as global finite element shape functions. However, as demonstrated below, the above formula is the key to build a partition of unity when the finite element τ is severed by a crack.

2.3 Element Partition Shape Functions: The Family \mathcal{F}_N^p

The construction of the so-called element partition shape functions is based on the following observation:

Let $\{L_i\}_{i \in \mathcal{I}}$ denote a set of functions which can approximate well, in an appropriate norm $\|\cdot\|_E$, the solution u of a boundary value problem, i.e., there exist u_i , $i \in \mathcal{I}$ such that

$$\|u_{hp} - u\|_E < \epsilon$$

where $u_{hp} = \sum_{i \in \mathcal{I}} u_i L_i$ and \mathcal{I} denotes an index set.

Now consider the following set of *cloud or element partition shape functions*, defined as

$$\phi_i^\alpha := \varphi_\alpha L_i, \quad \alpha = 1, \dots, N, \quad i \in \mathcal{I}$$

where φ_α is a partition of unity function. Then it is not difficult to show that linear combinations of these EP shape functions can also approximate well the function u

$$\begin{aligned} \sum_\alpha \sum_i u_i \phi_i^\alpha &= \sum_\alpha \sum_i u_i \varphi_\alpha L_i = \sum_\alpha \varphi_\alpha \sum_i u_i L_i \\ &= \sum_\alpha \varphi_\alpha u_{hp} = u_{hp} \sum_\alpha \varphi_\alpha = u_{hp} \end{aligned} \tag{2}$$

Note that:

- (i) The EP shape functions ϕ_i^α , $\alpha = 1, \dots, N$, $i \in \mathcal{I}$, are non-zero only over the cloud ω_α .
- (ii) Linear combination of the EP shape functions can reproduce the approximation u_{hp} to the function u .
- (iii) The functions L_i , $i \in \mathcal{I}$, can be chosen with great freedom. The most straightforward choice is polynomial functions since they can approximate well smooth functions. However, for many classes of problems, including the case of fracture mechanics problems, there are better choices. This case is discussed in details in Section 3.

In this section, element partition shape functions are defined in an n -dimensional setting using the same ideas outlined above.

Let the functions φ_α , $\alpha = 1, \dots, N$, denote a finite element partition of unity subordinate to the open covering $\mathcal{T}_N = \{\omega_\alpha\}_{\alpha=1}^N$ of a domain $\Omega \subset \mathbb{R}^n$, $n = 1, 2, 3$. Here, N is the number of vertex nodes in the finite element mesh. The cloud ω_α is the union of the finite elements sharing the vertex node \mathbf{x}_α .

Let $\chi_\alpha(\omega_\alpha) = \text{span}\{L_{i\alpha}\}_{i \in \mathcal{I}(\alpha)}$ denote local spaces defined on ω_α , $\alpha = 1, \dots, N$, where $\mathcal{I}(\alpha), \alpha = 1, \dots, N$, are index sets.

Suppose that the finite element shape functions, φ_α , are linear functions and that

$$\mathcal{P}_{p-1}(\omega_\alpha) \subset \chi_\alpha(\omega_\alpha) \quad \alpha = 1, \dots, N,$$

where \mathcal{P}_{p-1} denotes the space of polynomials of degree less or equal to $p - 1$. The element partition shape functions of degree p are defined by [33, 38]

$$\mathcal{F}_N^p = \{\varphi_i^\alpha = \varphi_\alpha L_{i\alpha}, \alpha = 1, \dots, N, i \in \mathcal{I}(\alpha)\} \quad (3)$$

Note that there is considerable freedom in the choice of the local spaces χ_α . The most obvious choice for a basis of χ_α is polynomial functions which can approximate well smooth functions. In this case, the EPM is essentially identical to the classical FEM. The implementation of hp adaptivity is, however, greatly simplified by the EP framework. Since each basis functions, $\{L_{i\alpha}\}_{i \in \mathcal{I}(\alpha)}$, $\alpha = 1, \dots, N$, can have a different polynomial order, we can have different polynomial orders associated with each vertex node of the finite element mesh [38]. The approximations can also be non-isotropic (i.e., different polynomial orders in different directions), regardless of the choice of finite element partition of unity (hexahedral, tetrahedral, etc.). The concept of edge and middle nodes, which is used in conventional p FEMs, is not needed in the framework of EPM. The implementation of h adaptivity is also simplified in EP methods since it needs to be done only on the partition of unity (linear finite elements in this case). Therefore, the implementation of h adaptivity for high-order approximations is the same as for linear approximations (there is no need, for example, of using high-order constraints as done in traditional hp finite element methods).

There are many situations in which the solution of a boundary value problem is not a smooth function. In these situations, the use of polynomials to build the approximation space, as in the FEM, may be far from optimal and may lead to poor approximations of the solution u unless carefully designed meshes are used. In the EPM, we can use any a-priori knowledge about the solution to make better choices for the local spaces χ_α . For example, in Section 13, we solve a boundary value problem in which the solution possesses point or lines of singularities at some parts of the domain. Then we can use the local spaces χ_α to build element partition shape functions that represent these singularities much more effectively than polynomial functions. The construction of these customized shape functions is discussed in details in Section 3.

Suppose that the partitions of Ω into finite elements which form the partition of unity satisfy the usual regularity assumptions in two and three dimensions, and, in one dimension, the ratio between the lengths of neighboring elements is bounded. We denote

$$\begin{aligned} h_\alpha &= \text{diam}(\omega_\alpha) \\ h &= \max_{\alpha=1, \dots, N} h_\alpha \end{aligned}$$

and

$$\mathbf{X}^{hp} = \text{span}\{\phi_i^\alpha\}, \quad \alpha = 1, \dots, N, \quad i \in \mathcal{I}(\alpha)$$

where the EP shape functions ϕ_i^α are defined in (3). In addition, suppose the unity function, 1, belongs to the set of local approximation functions, i.e.,

$$1 \in \chi_\alpha(\omega_\alpha) \quad \alpha = 1, \dots, N,$$

and that there exists a quantity ϵ depending of α, h, p , and u , such that

$$\|u - u_\alpha\|_{E(\Omega \cap \omega_\alpha)} \leq \epsilon(\alpha, h, p, u) \quad \alpha = 1, \dots, N$$

Then, it can be proved that [17, 33] there $\exists u_{hp} \in \mathbf{X}^{hp}$ such that

$$\|u - u_{hp}\|_{E(\Omega)} \leq C \left(\sum_{\alpha=1}^{N(h)} \epsilon^2(\alpha, h, p, u) \right)^{1/2}$$

where the constant C is independent of u, h, p .

2.4 Construction of Discontinuous Element Partition Shape functions

In this section, a technique to modify a finite element POU at elements cut by a crack surface is described. The POU is modified such that a discontinuity in the displacement field across the crack surface is created. The POU is modified only for elements cut by the crack. Elsewhere, a finite element partition of unity, as described in Section 2.1, is used. The technique allows for elements to be arbitrarily cut by the crack surface without any mesh modification. In these elements, the POU is built using the Shepard formula (1) and finite element shape functions as weighting functions. The technique is described in a two-dimensional setting in order to simplify the notation and pictures, however, the technique can be used without any modifications in the three dimensional case. Three-dimensional problems are solved in Section 13 using this technique. We also restrict ourselves to the case of quadrilateral elements. The case of triangular elements is identical.

The technique is described using the discretization depicted in Fig. 3. For elements that do not intersect the crack surface, the POU is provided by the finite element shape functions. For example, at any point $\mathbf{x} \in \tau_1$, there are four partition of unity functions

$$\varphi_1 = N_1, \quad \varphi_2 = N_2, \quad \varphi_5 = N_5, \quad \varphi_6 = N_6$$

Since the FE shape function form a POU, the following holds

$$\varphi_1(\mathbf{x}) + \varphi_2(\mathbf{x}) + \varphi_5(\mathbf{x}) + \varphi_6(\mathbf{x}) = N_1(\mathbf{x}) + N_2(\mathbf{x}) + N_5(\mathbf{x}) + N_6(\mathbf{x}) = 1 \quad \forall \mathbf{x} \in \tau_1$$

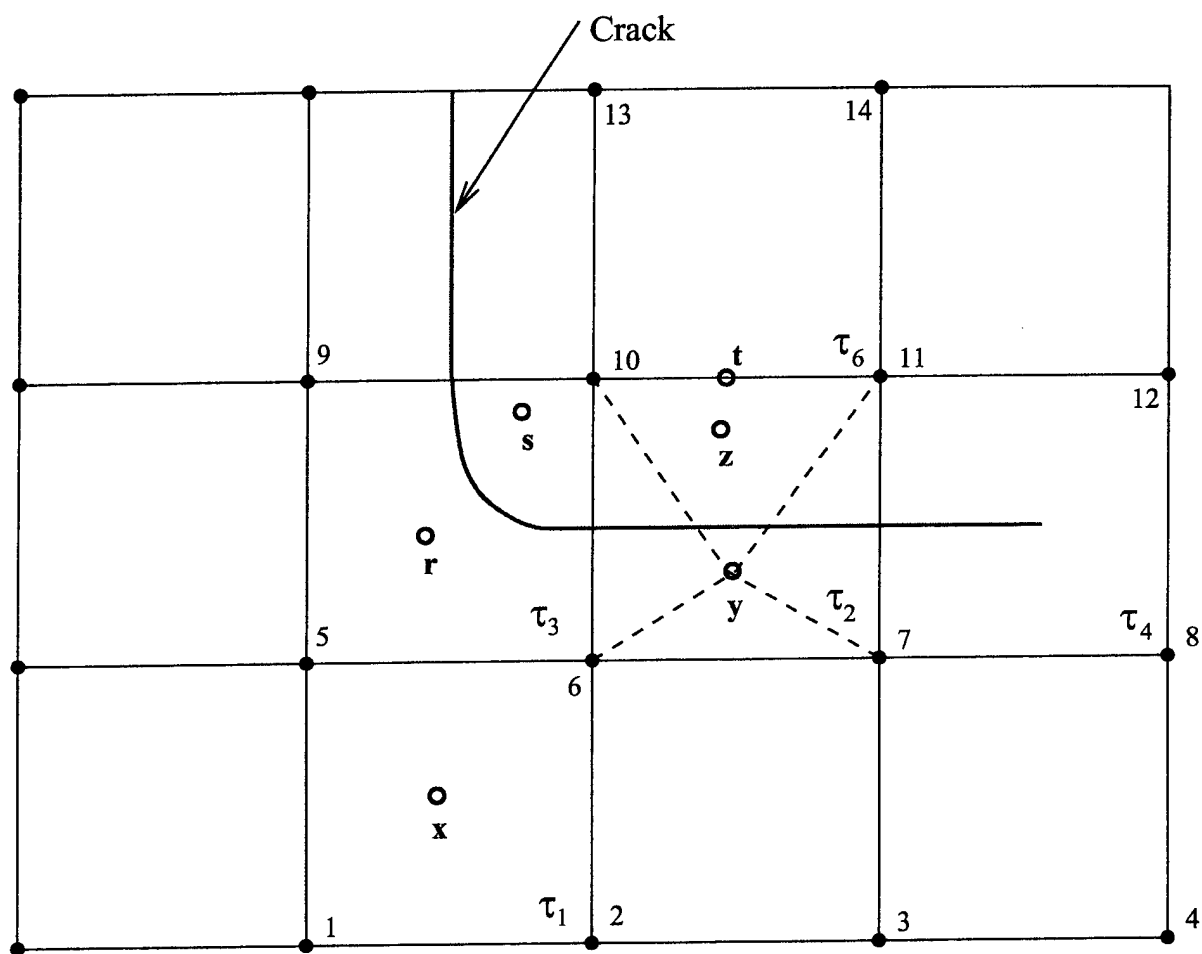


Figure 3: Example of element partition discretization cut by a crack

The Shepard formula (1) can also be used to build the POU but in the case of, for example, element τ_1 , the results, in this case, are trivial, i.e.,

$$\varphi_\alpha(\mathbf{x}) = \frac{N_\alpha(\mathbf{x})}{N_1(\mathbf{x}) + N_2(\mathbf{x}) + N_5(\mathbf{x}) + N_6(\mathbf{x})} = N_\alpha(\mathbf{x}) \quad \alpha = 1, 2, 5, 6$$

For the elements that are cut by the crack surface, the Shepard formula (1) is used in combination with the visibility approach [4]. That is, the crack surface is taken as an opaque surface and, at a given point $\mathbf{x} \in \tau$, the weighting function N_α is taken as non-zero if and only if the segment connecting \mathbf{x} and \mathbf{x}_α does not intersect the crack surface. As an illustration, consider the case of the element τ_2 shown in Fig. 3. At the point $\mathbf{y} \in \tau_2$, according to the visibility criteria, the only non-zero weighting functions are N_6 and N_7 , since the segments $[\mathbf{y} - \mathbf{x}_{10}]$ and $[\mathbf{y} - \mathbf{x}_{11}]$ intersect the crack surface. The partition of unity at $\mathbf{y} \in \tau_2$ is then given by

$$\varphi_6(\mathbf{y}) = \frac{N_6(\mathbf{y})}{N_6(\mathbf{y}) + N_7(\mathbf{y})} \quad \varphi_7(\mathbf{y}) = \frac{N_7(\mathbf{y})}{N_6(\mathbf{y}) + N_7(\mathbf{y})}$$

Note that

$$\varphi_6(\mathbf{y}) + \varphi_7(\mathbf{y}) = \frac{N_6(\mathbf{y}) + N_7(\mathbf{y})}{N_6(\mathbf{y}) + N_7(\mathbf{y})} = 1$$

Therefore, the functions φ_6 and φ_7 , as defined above, constitute a partition of unity.

At the point $\mathbf{z} \in \tau_2$, the partition of unity is given by

$$\varphi_{10}(\mathbf{z}) = \frac{N_{10}(\mathbf{z})}{N_{10}(\mathbf{z}) + N_{11}(\mathbf{z})} \quad \varphi_{11}(\mathbf{z}) = \frac{N_{11}(\mathbf{z})}{N_{10}(\mathbf{z}) + N_{11}(\mathbf{z})}$$

Therefore, *the approximation is discontinuous across the crack surface.*

As another example, consider the case of the element τ_3 with nodes $\mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_9$ and \mathbf{x}_{10} as depicted in Fig. 3. At the point $\mathbf{r} \in \tau_3$, the partition of unity is given by

$$\begin{aligned} \varphi_5(\mathbf{r}) &= \frac{N_5(\mathbf{r})}{N_5(\mathbf{r}) + N_6(\mathbf{r}) + N_9(\mathbf{r})} & \varphi_6(\mathbf{r}) &= \frac{N_6(\mathbf{r})}{N_5(\mathbf{r}) + N_6(\mathbf{r}) + N_9(\mathbf{r})} \\ \varphi_9(\mathbf{r}) &= \frac{N_9(\mathbf{r})}{N_5(\mathbf{r}) + N_6(\mathbf{r}) + N_9(\mathbf{r})} \end{aligned}$$

At the point $\mathbf{s} \in \tau_3$, the partition of unity is given by

$$\varphi_{10}(\mathbf{s}) = \frac{N_{10}(\mathbf{s})}{N_{10}(\mathbf{s})} = 1$$

Let us now show that the approximation is continuous at points not at the crack surface. Consider, for example, the point \mathbf{t} located at the boundary between elements τ_2 and τ_6 , as

shown in Fig. 3. Consider the POU function φ_{10} associated with node \mathbf{x}_{10} . If this function is computed from element τ_2 we have

$$\varphi_{10}^{(-)}(\mathbf{t}) = \frac{N_{10}(\mathbf{t})}{N_{10}(\mathbf{t}) + N_{11}(\mathbf{t})}$$

If the function φ_{10} is computed from element τ_6 we have

$$\varphi_{10}^{(+)}(\mathbf{t}) = \frac{N_{10}(\mathbf{t})}{N_{10}(\mathbf{t}) + N_{11}(\mathbf{t}) + N_{13}(\mathbf{t}) + N_{14}(\mathbf{t})} = \frac{N_{10}(\mathbf{t})}{N_{10}(\mathbf{t}) + N_{11}(\mathbf{t})}$$

Therefore $\varphi_{10}^{(-)}(\mathbf{t}) = \varphi_{10}^{(+)}(\mathbf{t})$ and the function φ_{10} is continuous at \mathbf{t} . This same argument can be used at other points not located at the crack surface.

At the elements intersecting the crack front, like element τ_4 shown in Fig. 3, the crack is modeled with the aid of singular functions as described in Section 3.

Note that the above technique allows arbitrary cut of the finite element mesh by the crack surface. Also, the computational cost of EP shape functions is basically the same as finite element shape functions (even for elements cut by the crack surface). The EP shape functions are much more computationally efficient than, e.g., moving least square functions which are used in several meshless methods. Moreover, all the computations can be carried out at the element level as in standard finite element codes. And, importantly, the numerical integration of the stiffness and mass matrices can be done as efficiently as in standard finite element computations since the intersections of the global EP shape functions coincide with the integration domains. This is in clear contrast with meshless methods based on moving least square functions where the integration of the stiffness and mass matrices is very computationally expensive.

Another way of introducing a discontinuity in the EP shape functions is by multiplying the partition of unity functions φ_α by basis functions $L_{i\alpha}$ that are discontinuous. From the definition of the EP shape functions given in (3) it is observed that if the functions $L_{i\alpha}$ are discontinuous, the resulting EP shape functions are also discontinuous. The construction of discontinuous functions $L_{i\alpha}$ that can approximate well the solution in the neighborhood of a 3-dimensional crack front is discussed in the next section. The modeling of a crack in the EPM is therefore done with the aid of the technique outlined in this section, which creates discontinuities in the shape functions of elements cut by a crack, and with the aid of singular functions near the crack front, which, in addition of rendering a discontinuous field at the crack front, also allows accurate approximation of the solution near the crack front without any mesh modifications.

3 Customized Shape Functions for an Edge in 3D

There are many classes of problems for which the structure of the underlying partial differential equation can be exploited. Oden and Duarte [37, 39] have demonstrated how knowledge of the solution of the elasticity equations near a corner in two dimensional space can be used in a partition of unity method to efficiently model the singularities that occur in this class of problems. This type of singularity is resolved very poorly by polynomial functions such as are used in traditional finite element methods, unless a very refined mesh is used. In this section, the formulation proposed by Oden and Duarte [37, 39] is extended to the case of edges in three-dimensional space. Numerical examples are presented in Section 13.

Consider an straight edge in three-dimensional space as depicted in Fig. 4. In the figure, $2\pi - \alpha$ is the opening angle. For example, in the case of a crack, $\alpha = 2\pi$. Associated with the edge/crack, there is a Cartesian local coordinate system (ξ, η, ζ) and a cylindrical coordinate system (r, θ, ζ') with origins at (O_x, O_y, O_z) .

The displacement field $\mathbf{u}(r, \theta, \zeta')$ in the neighborhood of the edge (for points far from its vertices) can be written as [46, 47]

$$\mathbf{u}(r, \theta, \zeta') = \begin{Bmatrix} u_\xi(r, \theta) \\ u_\eta(r, \theta) \\ u_\zeta(r, \theta) \end{Bmatrix} = \sum_{j=1}^M \left[A_j^{(1)} \begin{Bmatrix} u_{\xi j}^{(1)}(r, \theta) \\ u_{\eta j}^{(1)}(r, \theta) \\ 0 \end{Bmatrix} + A_j^{(2)} \begin{Bmatrix} u_{\xi j}^{(2)}(r, \theta) \\ u_{\eta j}^{(2)}(r, \theta) \\ 0 \end{Bmatrix} + \right. \\ \left. + A_j^{(3)} \begin{Bmatrix} 0 \\ 0 \\ u_{\zeta j}^{(3)}(r, \theta) \end{Bmatrix} \right] + \begin{Bmatrix} \underline{u}_\xi(r, \theta) \\ \underline{u}_\eta(r, \theta) \\ \underline{u}_\zeta(r, \theta) \end{Bmatrix} \quad (4)$$

where (r, θ, ζ') are the cylindrical coordinates relatives to the system shown in Fig. 4, $u_\xi(r, \theta)$, $u_\eta(r, \theta)$ and $u_\zeta(r, \theta)$ are Cartesian components of \mathbf{u} in the ξ -, η - and ζ - directions, respectively, $\underline{u}_\xi(r, \theta)$, $\underline{u}_\eta(r, \theta)$ and $\underline{u}_\zeta(r, \theta)$ are functions smoother than any term in the sum.

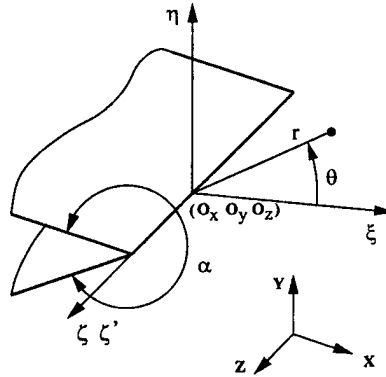


Figure 4: Coordinate systems associated with an edge in 3D space.

Assuming that the boundary is traction-free and neglecting body forces, the functions $u_{\xi j}^{(1)}$, $u_{\eta j}^{(1)}$, $u_{\xi j}^{(2)}$, $u_{\eta j}^{(2)}$ are given by [46, 47]

$$\begin{aligned} u_{\xi j}^{(1)}(r, \theta) &= \frac{r^{\lambda_j^{(1)}}}{2G} \{ [\kappa - Q_j^{(1)}(\lambda_j^{(1)} + 1)] \cos \lambda_j^{(1)} \theta - \lambda_j^{(1)} \cos(\lambda_j^{(1)} - 2)\theta \} \\ u_{\xi j}^{(2)}(r, \theta) &= \frac{r^{\lambda_j^{(2)}}}{2G} \{ [\kappa - Q_j^{(2)}(\lambda_j^{(2)} + 1)] \sin \lambda_j^{(2)} \theta - \lambda_j^{(2)} \sin(\lambda_j^{(2)} - 2)\theta \} \\ u_{\eta j}^{(1)}(r, \theta) &= \frac{r^{\lambda_j^{(1)}}}{2G} \{ [\kappa + Q_j^{(1)}(\lambda_j^{(1)} + 1)] \sin \lambda_j^{(1)} \theta + \lambda_j^{(1)} \sin(\lambda_j^{(1)} - 2)\theta \} \\ u_{\eta j}^{(2)}(r, \theta) &= -\frac{r^{\lambda_j^{(2)}}}{2G} \{ [\kappa + Q_j^{(2)}(\lambda_j^{(2)} + 1)] \cos \lambda_j^{(2)} \theta + \lambda_j^{(2)} \cos(\lambda_j^{(2)} - 2)\theta \} \end{aligned}$$

where the eigenvalues $\lambda_j^{(1)}$, $\lambda_j^{(2)}$ are found by solving

$$\begin{aligned} \sin \lambda_j^{(1)} \alpha + \lambda_j^{(1)} \sin \alpha &= 0 \\ \sin \lambda_j^{(2)} \alpha - \lambda_j^{(2)} \sin \alpha &= 0 \end{aligned}$$

In the case of a crack, where $\alpha = 2\pi$, the eigenvalues are $\lambda_j^{(1)} = \lambda_j^{(2)} = \lambda_j$

$$\lambda_1 = \frac{1}{2}, \quad \lambda_j = \frac{j+1}{2} \quad j \geq 2$$

The material constant κ and G are

$$\kappa = 3 - 4\nu \quad G = \frac{E}{2(1 + \nu)}$$

where E is the Young's modulus and ν is the Poisson's ratio.

The parameters $Q_j^{(1)}$ and $Q_j^{(2)}$ are given by

$$\begin{aligned} Q_j^{(1)} &= -\frac{\cos(\lambda_j^{(1)} - 1)\alpha/2}{\cos(\lambda_j^{(1)} + 1)\alpha/2} = -\Lambda_j^{(1)} \frac{\sin(\lambda_j^{(1)} - 1)\alpha/2}{\sin(\lambda_j^{(1)} + 1)\alpha/2} \\ Q_j^{(2)} &= -\frac{\sin(\lambda_j^{(2)} - 1)\alpha/2}{\sin(\lambda_j^{(2)} + 1)\alpha/2} = -\Lambda_j^{(2)} \frac{\cos(\lambda_j^{(2)} - 1)\alpha/2}{\cos(\lambda_j^{(2)} + 1)\alpha/2} \end{aligned}$$

where

$$\Lambda_j^{(s)} = \frac{\lambda_j^{(s)} - 1}{\lambda_j^{(s)} + 1} \quad s = 1, 2$$

In the case of a crack,

$$Q_j^{(1)} = \begin{cases} -1 & j = 3, 5, 7, \dots \\ -\Lambda_j^{(1)} & j = 1, 2, 4, 6, \dots \end{cases} \quad Q_j^{(2)} = \begin{cases} -1 & j = 1, 2, 4, 6, \dots \\ -\Lambda_j^{(2)} & j = 3, 5, 7, \dots \end{cases}$$

The functions $u_{\zeta j}^{(3)}$ are given by [46] (assuming that the boundary is traction-free and neglecting body forces)

$$u_{\zeta j}^{(3)} = \begin{cases} \frac{r_{\lambda_j^{(3)}}}{2G} \sin \lambda_j^{(3)} \theta & j = 1, 3, 5, \dots \\ \frac{r_{\lambda_j^{(3)}}}{2G} \cos \lambda_j^{(3)} \theta & j = 2, 4, 6, \dots \end{cases}$$

where

$$\lambda_j^{(3)} = \frac{j\pi}{\alpha} \quad j = 1, 2, \dots$$

Before using the above functions to build customized element partition shape functions, they have first to be transformed to the physical coordinates (x, y, z) as follows:

Define

$$\begin{aligned} \hat{u}_{\xi j}^{(s)}(\xi, \eta, \zeta) &= u_{\xi j}^{(s)} \circ \mathbf{T}_1^{-1}(\xi, \eta, \zeta) \\ \hat{u}_{\eta j}^{(s)}(\xi, \eta, \zeta) &= u_{\eta j}^{(s)} \circ \mathbf{T}_1^{-1}(\xi, \eta, \zeta) \quad s = 1, 2 \quad j = 1, \dots, M \\ \hat{u}_{\zeta j}^{(3)}(\xi, \eta, \zeta) &= u_{\zeta j}^{(3)} \circ \mathbf{T}_1^{-1}(\xi, \eta, \zeta) \end{aligned}$$

$$\mathbf{T}_1^{-1} : (\xi, \eta, \zeta) \rightarrow (r, \theta, \zeta') \quad (5)$$

$$\begin{Bmatrix} r \\ \theta \\ \zeta' \end{Bmatrix} = \begin{Bmatrix} \sqrt{\xi^2 + \eta^2} \\ \arctan(\frac{\eta}{\xi}) \\ \zeta \end{Bmatrix} \quad (6)$$

The coordinate system (ξ, η, ζ) is shown in Fig. 4.

Next define

$$\begin{aligned} \tilde{u}_{\xi j}^{(s)}(x, y, z) &= \hat{u}_{\xi j}^{(s)} \circ \mathbf{T}_2^{-1}(x, y, z) \\ \tilde{u}_{\eta j}^{(s)}(x, y, z) &= \hat{u}_{\eta j}^{(s)} \circ \mathbf{T}_2^{-1}(x, y, z) \quad s = 1, 2 \quad j = 1, \dots, M \\ \tilde{u}_{\zeta j}^{(3)}(x, y, z) &= \hat{u}_{\zeta j}^{(3)} \circ \mathbf{T}_2^{-1}(x, y, z) \end{aligned}$$

$$\mathbf{T}_2^{-1} : (x, y, z) \rightarrow (\xi, \eta, \zeta) \quad (7)$$

$$\begin{Bmatrix} \xi \\ \eta \\ \zeta \end{Bmatrix} = \mathbf{R}_2^{-1} \begin{Bmatrix} x - O_x \\ y - O_y \\ z - O_z \end{Bmatrix} \quad (8)$$

where $\mathbf{R}_2^{-1} \in \mathbb{R}^3 \times \mathbb{R}^3$, with rows given by the base vectors of the coordinate system (ξ, η, ζ) written with respect of the base vectors of the coordinate system (x, y, z) , and $\mathbf{O} = (O_x, O_y, O_z)$ are the coordinates of an arbitrary point along the edge (cf. Fig. 4). In the above, $\tilde{u}_{\xi j}^{(s)}(x, y, z)$,

$\tilde{u}_{\eta j}^{(s)}(x, y, z)$, $s = 1, 2$ and $\tilde{u}_{\zeta j}^{(3)}(x, y, z)$ are the components of the displacement vectors *in the directions* ξ , η and ζ , respectively, written in terms of the physical coordinates x , y and z . To get the components of \mathbf{u} in the directions x , y and z the following transformation needs to be applied:

$$\begin{Bmatrix} u_{xj}^{(s)}(\mathbf{x}) \\ u_{yj}^{(s)}(\mathbf{x}) \\ u_{zj}^{(3)}(\mathbf{x}) \end{Bmatrix} = \mathbf{R}_2 \begin{Bmatrix} \tilde{u}_{\xi j}^{(s)}(\mathbf{x}) \\ \tilde{u}_{\eta j}^{(s)}(\mathbf{x}) \\ \tilde{u}_{\zeta j}^{(s)}(\mathbf{x}) \end{Bmatrix} \quad s = 1, 2 \quad j = 1, \dots, M$$

where $\mathbf{R}_2 = (\mathbf{R}_2^{-1})^T$.

The construction of customized EP shape functions using singular functions as defined above, follows the same approach as in the case polynomial type shape functions. The singular functions are multiplied by the partition of unity functions φ_α associated with nodes near an edge/crack. In the computations of Section 13, the following singular functions are used in the construction of customized EP shape functions

$$u_{x1}^{(1)}, u_{y1}^{(1)}, u_{z1}^{(3)}, u_{x1}^{(2)}, u_{y1}^{(2)}, u_{z2}^{(3)}$$

which are given by

$$\begin{bmatrix} u_{x1}^{(1)} & u_{x1}^{(2)} \\ u_{y1}^{(1)} & u_{y1}^{(2)} \\ u_{z1}^{(3)} & u_{z2}^{(3)} \end{bmatrix} = \mathbf{R}_2 \begin{Bmatrix} \tilde{u}_{\xi 1}^{(1)} & \tilde{u}_{\xi 1}^{(2)} \\ \tilde{u}_{\eta 1}^{(1)} & \tilde{u}_{\eta 1}^{(2)} \\ \tilde{u}_{\zeta 1}^{(3)} & \tilde{u}_{\zeta 1}^{(3)} \end{Bmatrix}$$

The customized EP shape functions are then built as

$$\varphi_\alpha \times \{u_{x1}^{(1)}, u_{y1}^{(1)}, u_{z1}^{(3)}, u_{x1}^{(2)}, u_{y1}^{(2)}, u_{z2}^{(3)}\} \quad (9)$$

Here, α is the index of a finite element vertex node on or near an edge/crack in 3D.

4 A Posteriori Error Estimation for The EPM

Error estimation in PHLEXcrackTM is performed using an explicit residual technique. This method combines traditional residual error estimation techniques with methods based on interpolation theory. The resulting formulation for the error estimate can be written

$$\|u - u_{hp}\|_{E(\Omega)}^2 \leq 2NC_3^2 \sum_{k=1}^N \left[C_1^2 \|\bar{R}\|_{L_2(\Omega_k)}^2 + C_2^2 \|\bar{r}\|_{L_2(\Gamma_t \cap \partial\Omega_k)}^2 \right]$$

where $\|\cdot\|_E$ is the norm associated with the bilinear form, N is the number of domain subdivisions, Γ_t is the portion of $\partial\Omega$ with specified traction boundary conditions and \bar{R} and \bar{r} are the cloud-weighted interior and boundary residuals

$$\bar{R}^2 = R^2 \sum_{\omega_\alpha} [(hp)_\alpha^2]$$

$$\bar{r}^2 = r^2 \sum_{\omega_\alpha} [(hp)_\alpha]$$

with $(hp)_\alpha = h_\alpha/p_\alpha$ for points inside the cloud ω_α and zero elsewhere.

Without determining the constants C_1 , C_2 and C_3 we use the error indicators ϕ as the basis of adaptation and for determining convergence rates.

$$\phi_k^2 = \|\bar{R}\|_{L_2(\Omega_k)}^2 + \|\bar{r}\|_{L_2(\Gamma_t \cap \partial\Omega_k)}^2$$

5 Extraction of Stress Intensity Factors: The Least Squares Fit Method

Once the solution is obtained at some time step, the amount and direction of crack propagation over the next time increment can be determined. The crack front is represented as a series of straight line segments connected at vertices. The stress intensity factors (SIF's) are calculated at the vertex points along the crack front. Figure 5 represents the flowchart of the fracture dynamics process as implemented in PHLEXcrackTM.

The Least Squares Fit method is used for the calculations of SIF's. In this method, the SIF's are obtained by minimizing the errors among the discretized stresses calculated from the solution and the asymptotic values. The method has produced accurate results in finite element settings. In this project, it has been extended to be used with our three-dimensional dynamic CEPD model.

Define the least-squares functional as:

$$J(K_M^l(\mathbf{y})) := (\boldsymbol{\sigma}^h - \boldsymbol{\sigma}, \boldsymbol{\sigma}^h - \boldsymbol{\sigma})_{\mathbf{y}} \quad M = I - III \quad l = 1, \dots, l_{max}$$

where K_M^l is the l th stress intensity factor associated with mode M . The inner product $(\cdot, \cdot)_{\mathbf{y}}$ is defined as:

$$(\mathbf{u}, \mathbf{v})_{\mathbf{y}} = \sum_{\alpha=1}^N \left(\sum_{j=1}^m \sum_{i=1}^m u_i(\mathbf{x}_\alpha) D_{i,j}^{-1} v_j(\mathbf{x}_\alpha) \right) \mathcal{W}_\alpha(\mathbf{y}),$$

and (see Figure 6)

$\mathbf{u} = \{u_1, u_2, \dots, u_m\}$, $\mathbf{v} = \{v_1, v_2, \dots, v_m\}$ are any two vectors in \mathbb{R}^m ,

$\boldsymbol{\sigma}^h(\mathbf{x})$ is the discretized stress vector,

$\boldsymbol{\sigma}(\mathbf{x}) := \sum_{M=I}^{III} \sum_{l=1}^{l_{max}} [K_M^l \mathbf{F}_M^l(\mathbf{x})]$ is the asymptotic stress vector,

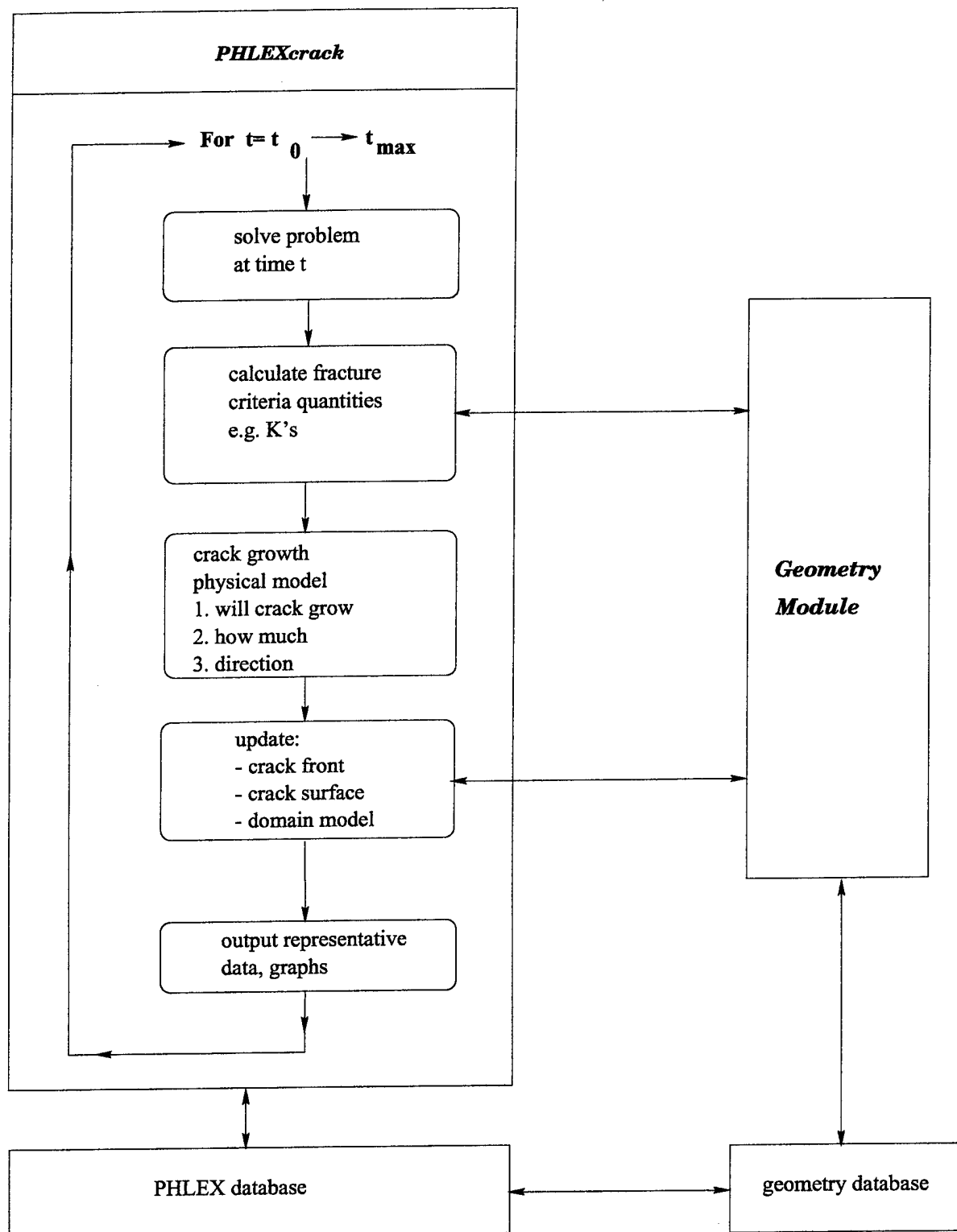


Figure 5: Flowchart of PHLEXcrackTM.

$\mathcal{W}_\alpha(\mathbf{y}) \in \mathbb{R}^n(\Omega)$ is a weighting function associated with sampling point \mathbf{x}_α given by

$$\mathcal{W}_\alpha(\mathbf{y}) = \frac{1}{\|\mathbf{y} - \mathbf{x}_\alpha\|_{R^n}^p} \quad \text{where } p \text{ is typically equal to 3-6.}$$

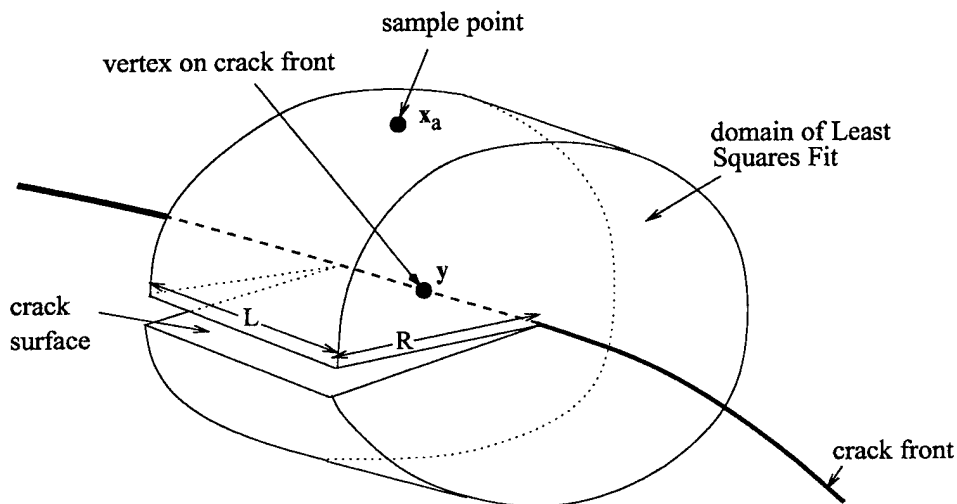


Figure 6: Domain used for the Least Square Fit method.

K_M^l s are found by minimizing the least-squares functional:

$$\frac{\partial J}{\partial K_M^l} = 0 \quad M = I - III \quad l = 1, 1 \dots, l_{max}$$

This leads to the following system of equations:

$$\sum_{M=I}^{III} \sum_{q=1}^{q_{max}} K_M^q(\mathbf{F}_M^q, \mathbf{F}_{M'}^l)_{\mathbf{y}} = (\boldsymbol{\sigma}^h, \mathbf{F}_{M'}^l)_{\mathbf{y}} \quad M' = I - III \quad l = 1, \dots, l_{max}$$

If only the first terms ($l = 1$) of the three modes are used, the three corresponding K 's are found by solving:

$$\begin{bmatrix} (\mathbf{F}_I^1, \mathbf{F}_I^1)_y & (\mathbf{F}_I^1, \mathbf{F}_{II}^1)_y & (\mathbf{F}_I^1, \mathbf{F}_{III}^1)_y \\ (\mathbf{F}_{II}^1, \mathbf{F}_I^1)_y & (\mathbf{F}_{II}^1, \mathbf{F}_{II}^1)_y & (\mathbf{F}_{II}^1, \mathbf{F}_{III}^1)_y \\ (\mathbf{F}_{III}^1, \mathbf{F}_I^1)_y & (\mathbf{F}_{III}^1, \mathbf{F}_{II}^1)_y & (\mathbf{F}_{III}^1, \mathbf{F}_{III}^1)_y \end{bmatrix} \begin{Bmatrix} K_I^1 \\ K_{II}^1 \\ K_{III}^1 \end{Bmatrix} = \begin{Bmatrix} (\boldsymbol{\sigma}^h, \mathbf{F}_I^1)_y \\ (\boldsymbol{\sigma}^h, \mathbf{F}_{II}^1)_y \\ (\boldsymbol{\sigma}^h, \mathbf{F}_{III}^1)_y \end{Bmatrix}$$

The domain used for the Least Squares Fit method is a cylinder centered at the vertex with its axis along the tangent of the crack front at that vertex. The dimensions of the cylinder and the number of sampling points in the r , θ , and z directions are input data. In addition, one can choose the type of the D matrix and the weight p .

In addition to the extraction method presented here and implemented in the current work, the user can develop and add his or her own method. This will be explained in Section 11.

6 Crack Evolution Models

The Stress Intensity Factors (SIF's) calculated above are used to determine whether the crack will advance or not, and the amount and direction of propagation, if any. The front is then advanced to its new position, and the numerical model updated accordingly.

Crack propagation quantities are calculated based on some physical models. Crack physics are not well known, particularly for three dimensional problems. Therefore, fracture models usually make extensive use of plane strain physics models. In this work, two physical models have been used. Moreover, the user can implement his/her own model as described in Section 11.

The Freund Model [20, 24, 44]:

Direction of crack growth in the plane normal to the crack front is given by:

$$\theta = 2 \tan^{-1} \left\{ \frac{1}{4} \left(\frac{K_I}{K_{II}} \pm \sqrt{\left(\frac{K_I}{K_{II}} \right)^2 + 8} \right) \right\} \quad (10)$$

for $K_{II} \neq 0$, and $\theta = 0$ for $K_{II} = 0$. In the equation above, θ is measured with respect to the forward vector \mathbf{n}_1 . Vector \mathbf{n}_1 is the crack front normal vector along the intersection of the plane normal to crack front with the plane tangent to it at the vertex (see Figure 7).

It was assumed that mode III does not affect crack direction; it only affects crack speed. The sign in equation (10) above is chosen to correspond to a positive stress intensity factor for the direction given by θ (positive if $K_{II} < 0$ and negative if $K_{II} > 0$). This corresponds to the direction normal to the maximum hoop stress.

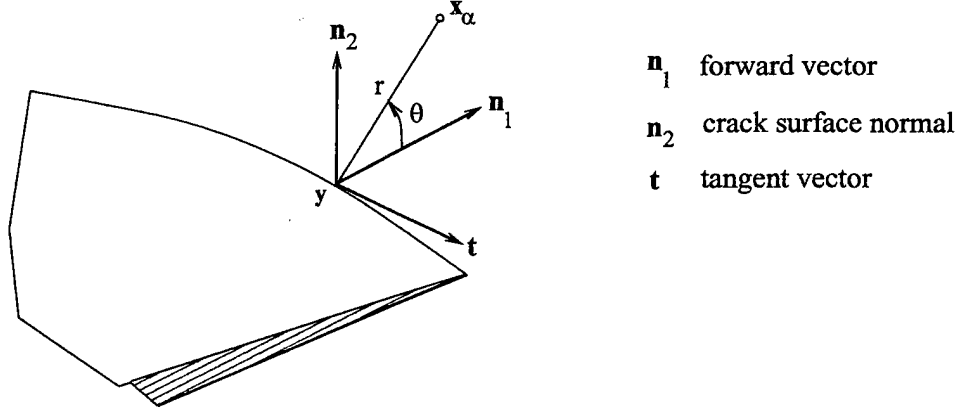


Figure 7: Local unit vectors at crack vertex.

The current energy release rate of a stationary crack is calculated at every vertex as:

$$G(0) = \frac{K_{I,equiv}^2}{E^*} + \frac{K_{III}^2}{2\mu},$$

where

$$K_{I,equiv} = K_I \cos^3(\theta/2) - \frac{3}{2} K_{II} \cos(\theta/2) \sin \theta,$$

μ is the shear modulus, and E^* is the effective Young's modulus. For plane strain, E^* is given by:

$$E^* = \frac{E}{1 - \nu^2}.$$

Crack will propagate at any vertex if:

$$G(0) > G_{crit}, \quad (11)$$

where G_{crit} is the critical energy release rate given by:

$$G_{crit} = \frac{K_{ID}^2(\dot{a})}{E^*} \approx \frac{\hat{K}_{ID}^2}{E^*}$$

and K_{ID} is the dynamic fracture toughness in a pure mode I crack; it is approximated by the constant value \hat{K}_{ID} (material property).

The speed at which the crack will propagate at a vertex (\dot{a}) is then calculated as the root of the quadratic equation:

$$\Psi \epsilon^2 - (\Psi + 1) \epsilon + \frac{\Gamma - 1}{\Gamma} = 0$$

where $\Gamma = G(0)/G_{crit} \geq 1$, $\Psi = c_R/c_{lim} > 1$, and $\epsilon = \dot{a}/c_R \leq 1$. c_{lim} is the limiting crack speed ($< c_R$) given as an input quantity. c_R is Rayleigh wave speed given as a root of:

$$4\beta_1\beta_2 - (1 + \beta_2^2)^2 = 0,$$

where,

$$\beta_1^2 = 1 - \left(\frac{c_R}{c_d}\right)^2 \quad (c_d \equiv \text{dilatational wave speed} = \sqrt{\frac{\kappa+1}{\kappa-1}} \sqrt{\frac{\mu}{\rho}} \text{ in plane strain}),$$

$$\beta_2^2 = 1 - \left(\frac{c_R}{c_s}\right)^2 \quad (c_s \equiv \text{shear wave speed} = \sqrt{\frac{\mu}{\rho}} \text{ in plane strain}), \text{ and}$$

$$\kappa \equiv \text{Kosolov constant} = 3 - 4\nu \text{ (for plane strain)}.$$

In the above, E is Young's modulus, μ is the shear modulus, ρ is the mass density, and ν is Poisson's ratio.

Prescribed Velocity Model:

In this model, the propagation angle is calculated using equation (10) above, and the propagation criteria is given by equation (11). The crack speed, however, is calculated according to a pre-defined function. This function is given as an input data. The function is usually a segmental linear function of time (refer to the User Manual).

7 Dynamic Flow-Structure Interaction Problems

7.1 Introduction

Although the scope of the project was restricted to the analysis of the solid structure in the dynamic regime (with the loads originated from the fluid-structure interaction), in this section we will present short overview of modern techniques used in this field. The present section will only provide a brief overview of the current state of the art and main contributions of the Computational Mechanics Company to these topics.

Computational Fluid Dynamics is presently dominated by finite volume and finite element techniques, which provide greater flexibility of geometry and mesh design than earlier finite difference methods. There exist a great number of general purpose and specialized CFD codes, covering flow ranges from low-speed non-Newtonian to hypersonic flow regimes. Their itemization is beyond the scope of this proposal, comprehensive lists can be found on the World Wide Web (<http://www.comco.com/feaworld>, http://icemcfd.com/cfd/CFD_codes.html).

COMCO has made significant and unique contributions to the development of finite element methods for CFD applications including, in particular, adaptive computational methods for CFD [23,36,40-43,53,54]. In fact, COMCO was the developer of the first hp-adaptive commercial CFD code P3/CFD marketed by PDA engineering in the early 1990's. More recently,

COMCO has developed a new, advanced hp-adaptive CFD code PHLEX_{cf}d and a specialized commercial extrusion modeling software HyperXtrude, marketed by Altair Engineering.

Flow-Structure Interaction (FSI) has been the topic of many research efforts, especially in application to hypersonics [49] and aeroelasticity [1,8,55]. Historically, most of the software development efforts in this area have been focused on selected applications and, accordingly, utilized some specific flow-structure coupling schemes (one-way, staggered one-way, two-way, etc.). To-date, virtually the only attempt at releasing a commercial general purpose multiphysics code is Spectrum(tm) by Centric - due to very complex problem setup and poor computational efficiency, it's success has been rather limited. It appears that developing specialized applications for specific types of flow-structure interaction is a more successful and efficient approach than general purpose applications.

In the area of flow interaction with deformable structures, COMCO has developed a unique module for modeling closely coupled dynamic flow-structure interactions. In particular, this module utilizes:

- large deflection structural model,
- arbitrary Lagrangian-Eulerian fluid solver, and
- a moving mesh algorithm,

to model fully coupled dynamic flow-structure interactions representative of flutter and buffeting. This application has been developed under sponsorship from Air Force Office of Scientific Research (AFOSR) [55].

COMCO has been developing computational software for various government installations and private industry for over a decade. A few of these development efforts which are relevant are as follows:

- – Title: Adaptive Computational Methods for Fluid Structure Interaction in Internal Flows
 - Description: This project focused on the development of an inviscid h-adaptive finite element compressible flow solver for modeling the transonic and supersonic flow through a jet engine. In particular, this effort used mesh sliding interface capabilities to move the rotor blade relative to the stator.
 - Client: NASA, contract NAS3-25196.
 - Contact: Jose Sans
 - Date of Completion: May 1989.
- – Title: Analysis of Flow-, Thermal- and Structural-Interaction of Hypersonic Structures Subjected to Severe Aerodynamic Heating

- Description: The project was focused on development of finite element techniques for coupling transient hypersonic flow simulation and heating rate prediction with thermal-structural response of hypersonic structures.
- Client: AFOSR
- Contact: Dr. Spencer Wu, (phone: 703-696-8523)
- Date of Completion: May 1990.
- – Title: STORESIM: An Integrated System for Multi-Body CFD Simulations Using Unstructured, Adaptive Grids.
 - Description: The primary objective of this project was to develop a state-of-the-art integrated software package to accurately and efficiently model the flow around the missiles, predict the trajectories by incorporating the fluid-structure interaction during missile deployment.
 - Client: Wright-Patterson Air Force Base
 - Contact: Frank C. Witzeman (phone: 937-255-3778, e-mail: witzeman@ind4.fim.wpafb.af.mil)
 - Date of Completion: December 1996.
- – Title: Dynamic Instabilities in Flow-Structure Interactions, Flutter and Buffeting
 - Description: The project was focused on a detailed study of dynamic instabilities in fluid-structure interaction and development of new computational techniques for the numerical simulation of a class of physical instabilities in which a viscous compressible fluid interacts with a flexible elastic structure undergoing large deformations.
 - Client: AFOSR
 - Contact: Mjr. Brian Sanders (phone: 703-696-7259, e-mail: brian.sanders@afosr.af.mil)
 - Date of Completion: December 1997.

The basics of linear flow-structure interactions have been covered in many monographs (see for example [7, 12]) and numerous scientific papers. Presently, most of the methods in applied aeroelasticity rely strongly on linearization of the equations of motion and simplified computational methods, such as analytical solutions, vortex lattice and doublet lattice models, or finite volume discretizations.

In the design of modern aerospace systems, the necessity of considering more complex, nonlinear physical phenomena in fluid structure interactions, such as those encountered in buffeting, fluid thermal structural interactions, and viscous turbulent boundary layer effects has pushed the analysis problem far beyond the limits of the classical theory and into a realm of phenomena for which a list of open physical, mathematical and computational issues exist. These problems require a fully nonlinear formulation of flow-structure interaction problems, a

more general notion of stability, applicable to nonlinear equations characterizing viscous flow interacting with a complex structure undergoing large deformations, and, possibly, large transient thermal gradients. The linearized stability elements of the classical theory are inadequate and new approaches to these classes of problems are needed.

Recently, several efforts have been reported towards expanding the flow-structure interaction solution into the realm of nonlinear flow analysis coupled in some way with structural deformations. Bendiksen [6] coupled a finite difference Euler CFD scheme with a simplified finite element structural representation to solve transient flow-structure interaction problems. Alonso et.al [1], Blackburn et.al. [8] and Morton et.al. [34] studied experimentally and numerically the unstable oscillation of a rigid cylinder induced by interaction with a uniform flow. Flutter of a nonlinear buckled plate was studied by Dowell [11] and Virgin and Dowell [56]. Various nonlinear aspects of aeroelasticity were discussed by Dugundji [18].

7.2 A General Problem Statement

This section presents a detailed formulation of dynamic flow-structure interaction problems and computational techniques used to solve them. This formulation was developed during the “Flutter and Buffeting” project mentioned above:

- it is applicable to fully coupled flow-structure interaction problems, with complete preservation of dynamic coupling terms on the interface,
- it can correctly represent large structural deflections and their interaction with the flow-field,
- it captures the nonlinearities that contribute to instabilities and define the final motion of the system (such as limit cycle oscillations), and
- it lends itself to finite element discretization or other numerical approximation methods.

The formulation combines an Arbitrary Lagrangian-Eulerian (ALE) approach for the fluid motion with an Updated Lagrangian Reference (ULR) description for the solid. The general idea and details of this formulation are presented in the remainder of this section.

A general illustration of the problem under consideration is shown in Fig. 8. Consider a solid body with a stress-free configuration Ω_0 . At the time t , the solid occupies a configuration Ω_t . In this configuration the solid is interacting with fluid, with a finite element or other mesh discretizing the flow domain. Because of the flow-structure interaction or other forces, the solid is moving towards the new current configuration Ω_{t+dt} . Accordingly, the mesh discretizing the fluid domain is also moving so that the boundary of the flow domain continues to match the new position of the solid. The mesh position is updated at every time step.

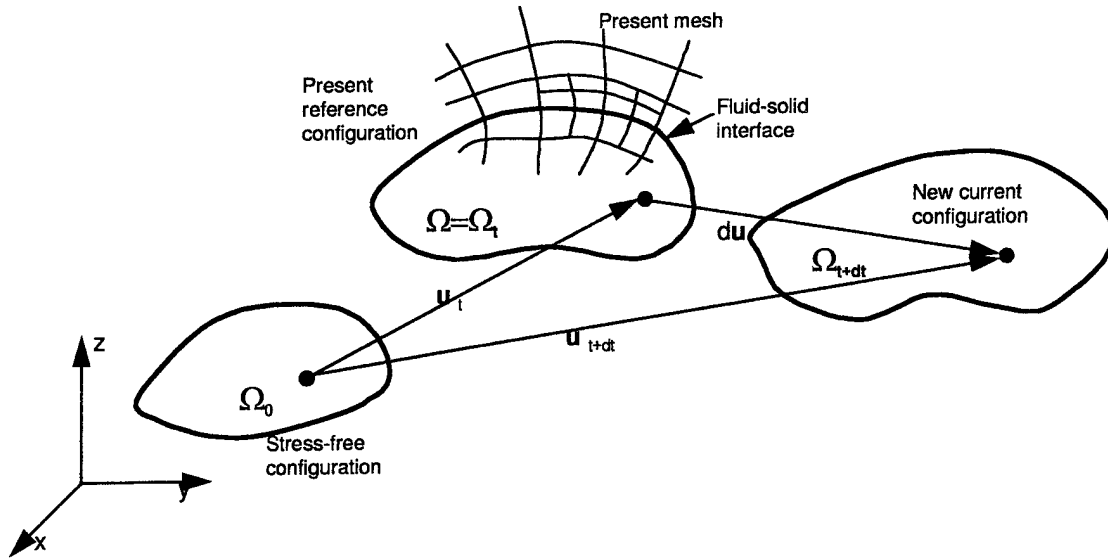


Figure 8: A general statement of the flow-structure interaction problem.

The details of the corresponding equations for the solid, fluid and the coupled problem are presented below.

7.3 Structural Deformation Problem

This section presents the formulation of the structural deformation problem. The main objective is to develop a formulation that is applicable to large deflection structural problems and is convenient for close coupling with flow equations and moving computational meshes.

The main assumptions considered for the structural problem are as follows:

- large deformations and small strains, and
- elastic material behavior.

The approach to the description of this problem is illustrated in Fig. 9. All the positions and displacements are represented in an inertial coordinate system x_i , $i = 1, 2, 3$. At some initial time $\tau = t_0$ the body occupies an initial, stress-free configuration Ω_0 . For a particle P , the position vector in this configuration is defined by a vector \mathbf{x}_0 . At a certain time $\tau = t$ the body is in a configuration Ω_t , and a position vector for a particle P is defined as \mathbf{x} . Typically, in the present solution process, this is the configuration reached in a transient

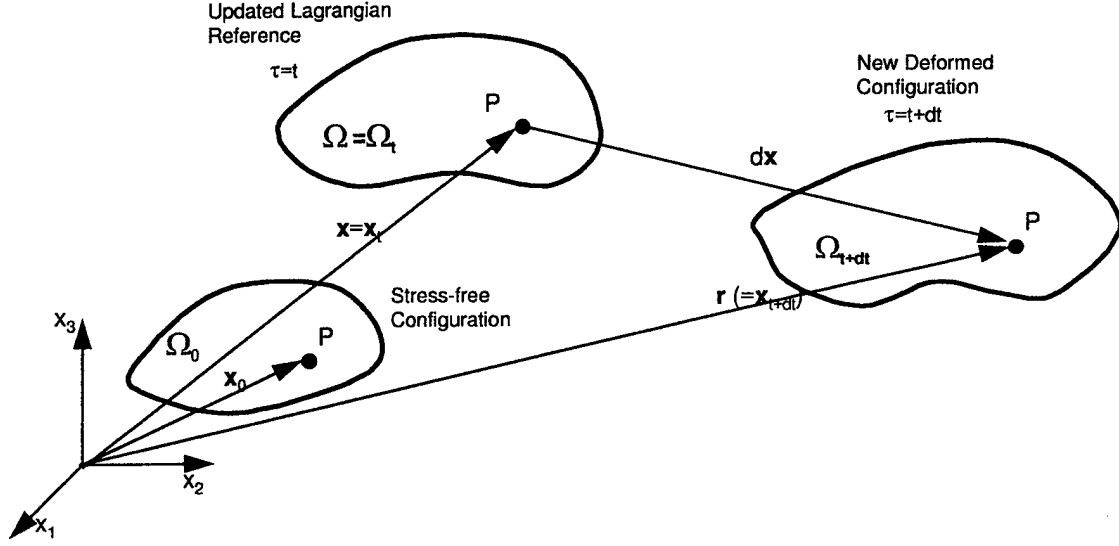


Figure 9: Transient structural deformation at large deformations.

solution process. At the most current time $\tau = t + dt$ the body is in a configuration Ω_{t+dt} , with the position vector for a particle P defined as \mathbf{r} . Typically, this is the configuration being calculated in a transient solution process.

Generally, the deformation can be viewed from an arbitrary reference configuration Ω , which need not coincide with any of the positions of the body during the motion. In the present flow-structure interaction problem, the reference configuration is chosen to be wherever the computational mesh is located. Typically, this will be the latest converged configuration Ω_t , as shown in Fig. 9. Importantly, all the differentiation and integration discussed subsequently will be performed on this reference configuration.

It should be noted that the reference configuration Ω is a Lagrangian reference, serving as a convenient viewpoint on the total deformation from Ω_0 to Ω_{t+dt} . As the solution progresses and the mesh moves, different configurations will be chosen as this reference frame. Thus, the present formulation was named an Updated Lagrangian Reference (ULR) formulation. Importantly, it differs from the Updated Lagrangian formulation in this that the total strains and stresses do not get accumulated during the computational process, but are always recalculated from the total deformation. Thus, accumulation of errors is avoided.

Within the above framework, the solution of a transient deformation problem can be stated as: *Knowing the present updated Lagrangian reference configuration Ω_t and the loads acting on the body, calculate the new current configuration Ω_{t+dt} after a time interval dt .*

Equations of Motion. Starting at this point, in order to differentiate between the solid and fluid domain, the domain corresponding to the solid body will be indicated by a superscript b . The equations of motion for the structure, expressed in the reference configuration Ω^B , are of the form:

$$\rho_B \mathbf{a}(\tau) = \text{div}_x \mathbf{T}(\tau) + \rho_B \mathbf{b}(\tau) \quad \text{in } \Omega^B \times [0, T] \quad (12)$$

where ρ_B is the density of the body in the reference configuration, τ is the time, \mathbf{a} is an acceleration vector, \mathbf{T} is a first Piola-Kirchhoff stress tensor, \mathbf{b} is a body force vector per unit mass and div_x is a divergence operator with respect to the reference configuration.

The equations of motion can also be expressed in terms of the second, symmetric Piola-Kirchhoff stress tensor as:

$$\rho_B \mathbf{a} = \text{div}_x (\mathbf{F} \mathbf{S}) + \rho_B \mathbf{b} \quad (13)$$

where the time symbol was dropped for clarity, \mathbf{S} denotes the second Piola Kirchhoff stress tensor and \mathbf{F} is a deformation gradient with respect to the reference configuration Ω^B :

$$\mathbf{F} = \text{grad}_x \mathbf{r} \quad (14)$$

Constitutive Equations. For the materials considered in this project, the constitutive equations are defined by a linear relationship between the Cauchy stress and strain tensors. It can be shown, using standard arguments of Continuum Mechanics [22, 49], that for large deflections / small strains problems this is equivalent to a linear relationship between the second Piola-Kirchhoff stress tensor and a Green-Lagrange strain tensor:

$$\mathbf{S} = \mathbf{C} \mathbf{E} \quad (15)$$

where \mathbf{C} is a fourth order elasticity tensor, satisfying the standard conditions of boundedness, symmetry and ellipticity. For isotropic materials, this can be expressed as:

$$\mathbf{S} = \lambda \text{tr}(\mathbf{E}) \mathbf{I} + 2\mu \mathbf{E} \quad (16)$$

where λ and μ are Lamé constants and \mathbf{I} is an identity tensor.

Kinematics and Strain Measures. In classical continuum mechanics, the Green strain tensor is defined as:

$$\mathbf{E} = \frac{1}{2} (\mathbf{F}^T \mathbf{F} - \mathbf{I}) \quad (17)$$

where \mathbf{F} is a deformation gradient from the stress-free configuration to the deformed configuration. This formula is valid under the usual assumption that the reference configuration coincides with the initial, stress-free configuration.

However, for the ULR description introduced above this is not the case, and a more general formula must be used. Considering that the Green strain tensor measures the change of metrics between the initial and final configuration as “seen” from the reference configuration [52], a generalized formula for the strain tensor in the ULR description is defined as:

$$\mathbf{E} = \frac{1}{2} (\mathbf{F}^T \mathbf{F} - \mathbf{F}_0^T \mathbf{F}_0) \quad (18)$$

where \mathbf{F}_0 is the deformation gradient from the reference configuration to the initial, stress-free configuration:

$$\mathbf{F}_0 = \mathbf{grad}_x \mathbf{x}_0 \quad (19)$$

It can be easily verified that the formula (18) reduces to the Green strain tensor or Almansi-Hamel strain tensor if the reference configuration coincides with the initial or the deformed configurations, respectively.

Boundary Conditions. For the above problem, the typical boundary conditions include: the prescribed motion on a part of the boundary, prescribed traction history, or their combinations. The prescribed motion is defined as:

$$\mathbf{r}(\mathbf{x}, \tau) = \hat{\mathbf{r}}(\mathbf{x}, \tau) \quad \text{on } \Gamma_u \times [0, T] \quad (20)$$

where Γ_u is the portion of the boundary with a prescribed displacement history and $\hat{\mathbf{r}}$ is the prescribed motion vector.

The boundary tractions $\mathbf{t}(\mathbf{x}, \tau)$ on the solid surface are expressed in terms of the first Piola-Kirchhoff stress tensor as:

$$\mathbf{t} = \mathbf{T} \mathbf{n} \quad (21)$$

where \mathbf{n} is a vector normal to the boundary in the reference configuration. Since this is the configuration discretized by a finite element mesh, the calculation of boundary normals is straightforward. For a portion of the boundary Γ_t , a traction history is prescribed as:

$$\mathbf{t}(\mathbf{x}, \tau) = \hat{\mathbf{t}}(\mathbf{x}, \tau) \quad \text{on } \Gamma_t \times [0, T] \quad (22)$$

where $\hat{\mathbf{t}}$ is the prescribed traction vector. On the flow-structure interface Γ^I , the tractions on the solid surface are a result of fluid pressure and viscous shear. The details of these interactions are presented later in this document.

Weak Formulation. Using a standard approach, namely multiplying (13) by a test function and integrating over the reference domain Ω^B , the following weak formulation of the equations of motion is obtained:

Find a mapping $\mathbf{r}(\tau)$ from $[0, T]$ to V such that:

$$\begin{aligned} \int_{\Omega^B} \rho_B \mathbf{w} \cdot \mathbf{a} \, d\Omega + \int_{\Omega^B} (\mathbf{grad}_x \mathbf{w})^T : (\mathbf{FS}) \, d\Omega = \\ \int_{\Omega^B} \rho_B \mathbf{w} \cdot \mathbf{b} \, d\Omega + \int_{\Gamma_t} \mathbf{w} \cdot \hat{\mathbf{t}} \, d\Gamma + \int_{\Gamma^I} \mathbf{w} \cdot (\mathbf{FS}) \mathbf{n}^B \, d\Gamma \end{aligned} \quad (23)$$

where $V = \{\mathbf{w} \in (H^1(\Omega^B))^N, (\mathbf{w}) = 0 \text{ on } \Gamma_u\}$ is the space of admissible displacements for the structural domain, with N denoting the dimensionality of the space ($N = 2$ or 3). In the above formulation, the Dirichlet boundary conditions on Γ_u are a part of the definition of the space V – in practice they are enforced through a penalty method or via direct matrix elimination. At this point, we will leave the boundary terms on the flow-structure interface Γ^I in the above generic form – the handling of flow-structure interactions will be discussed later.

7.4 Flow Equations; an ALE Description

This section presents flow equations in Arbitrary Lagrangian-Eulerian description, which utilizes a moving reference frame (grid) to handle changes in the flow domain due to structural motion.

Governing Equations. The transient incompressible flow in an ALE description is governed by the momentum equations:

$$\rho_F \dot{\mathbf{u}} + \rho_F ((\mathbf{u} - \mathbf{u}^g) \cdot \mathbf{grad}) \mathbf{u} - \text{div } \boldsymbol{\sigma} = \mathbf{f} \quad \text{in } \Omega^F \times [0, T] \quad (24)$$

augmented by the incompressibility condition:

$$\text{div } \mathbf{u} = 0 \quad (25)$$

Here ρ_F is the density of the fluid, \mathbf{u} is the flow velocity, \mathbf{u}^g is a prescribed grid velocity, $\boldsymbol{\sigma}$ is the stress tensor, \mathbf{f} represents body forces acting on the fluid, and Ω^F denotes the flow domain.

The grid velocity in the above equation is a prescribed field $\mathbf{u}^g(\mathbf{x}, \tau)$, which is generally quite arbitrary. In the approach adopted in this work, the grid velocity on the fluid-solid interface is equal to the velocity of the solid body, so that the grid follows the motion of the body.

The stress tensor in the fluid is expressed as:

$$\boldsymbol{\sigma} = -p \mathbf{I} + \nu(\text{grad} \mathbf{u} + (\text{grad} \mathbf{u})^T) \quad (26)$$

where p is the hydrostatic pressure and ν is fluid viscosity.

In numerical simulations it is often convenient to relax the incompressibility constraint and introduce a penalty approach to enforce its approximate satisfaction. In such cases, the pressure is expressed as:

$$p = -\lambda \text{div} \mathbf{u} \quad (27)$$

where λ is a (large) penalty number.

The typical boundary conditions for the flow domain include prescribed velocity on the inflow:

$$\mathbf{u}(\mathbf{x}, \tau) = \hat{\mathbf{u}}(\mathbf{x}, \tau) \quad \text{on } \Gamma_i \times [0, T] \quad (28)$$

prescribed traction (usually dominated by pressure):

$$\mathbf{t}(\mathbf{x}, \tau) = \hat{\mathbf{t}}(\mathbf{x}, \tau) \quad \text{on } \Gamma_p \times [0, T] \quad (29)$$

or solid wall no-slip boundary condition:

$$\mathbf{u} - \mathbf{u}^g = 0 \quad \text{on } \Gamma_s \times [0, T]. \quad (30)$$

Weak Formulation. Quite similarly as for the solid domain, the weak formulation for the flow equations is obtained by multiplying the governing equations by a test function and integrating over the domain of the fluid. After integrating relevant terms by parts, the final weak statement becomes:

Find a mapping (\mathbf{u}, p) from $[0, T]$ to $V \times Q$ such that:

$$\begin{aligned} (\rho_F \dot{\mathbf{u}}, \mathbf{w}) + c(\mathbf{u}, \mathbf{u}, \mathbf{w}) + a(\mathbf{u}, \mathbf{w}) + b(\mathbf{w}, p) &= (\mathbf{f}, \mathbf{w}) + b.i. \\ b(\mathbf{u}, q) &= 0 \end{aligned} \quad (31)$$

for every (\mathbf{w}, q) in $V \times Q$. The consecutive terms are defined as:

$$\begin{aligned} (\rho_F \dot{\mathbf{u}}, \mathbf{w}) &= \int_{\Omega^F} \rho_F \dot{\mathbf{u}} \cdot \mathbf{w} \, d\Omega \\ c(\mathbf{u}, \mathbf{u}, \mathbf{w}) &= \int_{\Omega^F} \rho_F \mathbf{u} \cdot \mathbf{grad} \, \mathbf{u} \cdot \mathbf{w} \, d\Omega \\ a(\mathbf{u}, \mathbf{w}) &= \int_{\Omega^F} \nu (\mathbf{grad} \, \mathbf{u} + (\mathbf{grad} \, \mathbf{u})^T) : \mathbf{grad} \, \mathbf{w} \, d\Omega \\ b(\mathbf{w}, p) &= - \int_{\Omega^F} \mathbf{div} \, \mathbf{w} \, p \, d\Omega \end{aligned}$$

and the boundary contributions denoted as *b.i.* are expressed as:

$$\int_{\Gamma} [\nu (\mathbf{grad} \, \mathbf{u} + (\mathbf{grad} \, \mathbf{u})^T) \mathbf{n}^F - p \mathbf{n}^F] \cdot \mathbf{w} \, d\Gamma$$

over the entire boundary of the flow domain (including the solid-fluid interface Γ^I). In the above, V and Q are the spaces of admissible flow velocities and pressures: $V = \{\mathbf{w} \in (H^1(\Omega))^N, (\mathbf{w}) = 0 \text{ on } \Gamma_i\}$ and $Q = \{q \in L^2(\Omega)\}$, respectively. The traction boundary conditions are applied by prescribing the value of the traction in the boundary integral:

$$\int_{\Gamma_p} \hat{\mathbf{t}} \cdot \mathbf{w} \, d\Gamma$$

and the Dirichlet boundary conditions on the inflow and solid wall are usually enforced via penalty or direct matrix operations.

7.5 Flow-Structure Interaction Equations

This section addresses the formulation of the dynamic flow-structure interaction problem at large structural deflections. The main objectives of this formulation are:

- to facilitate a fully coupled solution of the structural and the flow problem in a single time marching procedure, and
- to handle the dynamic interaction between the solid and the fluid with minimum computational cost or complexity.

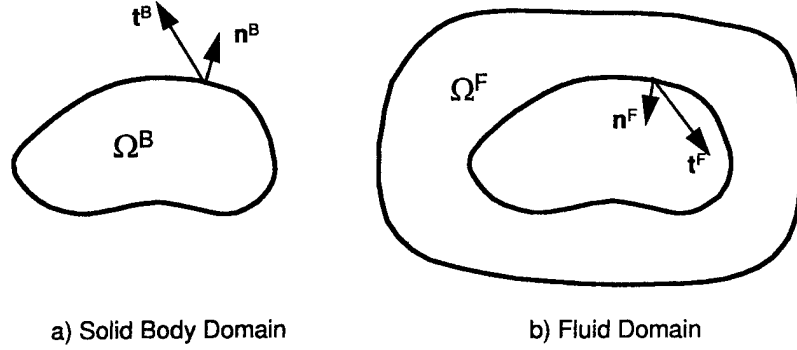


Figure 10: Fluid-structure interaction: complementary subdomains and interface conditions.

The structural deformation and fluid equations formulated so far are essentially disjoint, in the sense that they describe the behavior of the fluid and the solid separately, without any implication of interaction between these media. However, several characteristics of these formulations facilitate their application to the solution of a fully coupled flow-structure interaction problem, namely:

1. By selecting the reference configuration for the ULR structural description to coincide with the most recent configuration attained during the motion, the weak formulations for the structural and flow problem span complementary subdomains (see Fig. 10):

$$\Omega = \Omega^B \cup \Omega^F.$$

Therefore, a single mesh covering both the fluid and solid computational domains can be used in numerical simulations. Note that this would not be the case if the total Lagrangian description were used for the solid – the fixed solid computational domain would be “lagging” behind the changing flow domain.

2. Owing to the ALE formulation for the fluid domain, the grid motion can be used to continuously match the deformation of the solid.
3. Both the structural and flow equations represent momentum balance in true physical units. Therefore, the weak forms are compatible in the sense that the terms of each weak form represent virtual power and can be added together.

In the context of interaction problems, of special interest are the integrals on the flow-structure interface Γ^I . It can be easily observed that the corresponding boundary terms for the structural problem represent surface tractions on the solid body (see Fig. 10):

$$\int_{\Gamma^I} \mathbf{w} \cdot (\mathbf{F}\mathbf{S})\mathbf{n}^B \, d\Gamma = \int_{\Gamma^I} \mathbf{w} \cdot \mathbf{t}^B \, d\Gamma \quad (32)$$

and the similar also holds for the flow equations:

$$\int_{\Gamma^I} \mathbf{w} \cdot [\nu(\mathbf{grad} \mathbf{u} + (\mathbf{grad} \mathbf{u})^T) \mathbf{n}^F - p \mathbf{n}^F] d\Gamma = \int_{\Gamma^I} \mathbf{w} \cdot \mathbf{t}^F d\Gamma. \quad (33)$$

The interface conditions for the flow-structure interaction problem require that the motion of the grid matches the motion of the solid body:

$$\mathbf{u}^g = \dot{\mathbf{r}} \quad \text{on } \Gamma^I \quad (34)$$

and that the fluxes (tractions) cancel out across the interface:

$$\mathbf{t}^B = -\mathbf{t}^F \quad \text{on } \Gamma^I. \quad (35)$$

In the context of the weak statement, the total integral on the flow-structure interface consists of both fluid and solid contributions. In the light of equations (32) and (33), it amounts to:

$$\int_{\Gamma^I} \mathbf{w} \cdot (\mathbf{t}^B + \mathbf{t}^F) d\Gamma \quad (36)$$

which, because of the interface condition (35), is equal to zero. This means that in a solution scheme that utilizes the formulation presented above to simultaneously solve the union of a solid and a fluid domain, *there is no need to evaluate additional interface integrals* – the interface terms cancel out. This is subject to additional conditions, such as correctly calculated grid motion and using time integration schemes which do not introduce additional boundary terms as a by-product of the algorithm.

Also, it should be noted that the primary variables in the structural and fluid weak statements are different (positions vs. velocities), which complicates the direct formulation of a single problem encompassing both the fluid and the solid subdomains. This situation will be rectified by selecting appropriate time discretization schemes, with velocity becoming the primary variable for the entire domain. The details are discussed in the next section.

8 Solution of the Flow-Structure Interaction Problem

In order to solve the flow-structure interaction problem without introducing time lag error typical of staggered solution sequences, the approach developed in this project includes the entire domain in the computational scheme and handles the solid and the flow equations simultaneously. To facilitate this, appropriate versions of time marching algorithms have been introduced in which velocities are the primary unknowns for both the solid and fluid domains. In particular, in this work we used:

- for the solid domain, a one-step Newmark algorithm in velocity formulation, and
- for the flow domain, a one-step θ -method.

Based on the comments from the previous section, we expect the interface integrals to cancel out, to simplify the computational procedure. Additionally, a mesh movement algorithm will be used to continually update the computational domain. The basic steps and formulas of the numerical solution are discussed further in this section.

8.1 Time Integration for the Structural Domain

To advance the solution in time on the structural subdomain, a Newmark family of algorithms was used. The basic formulas of the Newmark family express positions and velocities at the time step $n + 1$ as:

$$\begin{aligned}\mathbf{x}^{n+1} &= \mathbf{x}^n + dt\mathbf{v}^n + \frac{dt^2}{2}[(1 - 2\beta)\mathbf{a}^n + 2\beta\mathbf{a}^{n+1}] \\ \mathbf{v}^{n+1} &= \mathbf{v}^n + (1 - \gamma)dt\mathbf{a}^n + \gamma dt\mathbf{a}^{n+1}\end{aligned}$$

where the index n indicates the previous, converged time step. β and γ are Newmark parameters, which affect dissipation and dispersion properties of the algorithm. In order to formulate a velocity version of this algorithm, the above formulas were recast to represent new positions and accelerations in terms of velocities:

$$\begin{aligned}\mathbf{x}^{n+1} &= \mathbf{x}^n + dt\left[\frac{\beta}{\gamma}(\mathbf{v}^{n+1} - \mathbf{v}^n) + \mathbf{v}^n\right] + \frac{dt^2}{2}\mathbf{a}^n\left(1 - \frac{2\beta}{\gamma}\right) \\ \mathbf{a}^{n+1} &= \frac{1}{\gamma dt}(\mathbf{v}^{n+1} - \mathbf{v}^n) - \frac{1 - \gamma}{\gamma}\mathbf{a}^n\end{aligned}$$

Substitution of these formulas to the weak statement for the solid leads to the following formulation:

$$\begin{aligned}\int_{\Omega^B} \frac{\rho^B}{\gamma dt} \mathbf{w} \cdot \mathbf{v}^{n+1} d\Omega + \int_{\Omega^B} (\mathbf{grad}_x \mathbf{w})^T : (\mathbf{F}^{n+1} \mathbf{S}^{n+1}) d\Omega = \\ \int_{\Omega^B} \rho_B \mathbf{w} \cdot \left(\frac{1}{\gamma dt} \mathbf{v}^n + \frac{1 - \gamma}{\gamma} \mathbf{a}^n\right) d\Omega + \int_{\Omega^B} \rho_B \mathbf{w} \cdot \mathbf{b}^{n+1} d\Omega + \int_{\Gamma_t} \mathbf{w} \cdot \hat{\mathbf{t}}^{n+1} d\Gamma\end{aligned}\tag{37}$$

This is an equation for calculating the velocities at a new time step t^{n+1} . Due to the implicit dependence of the nonlinear terms $\mathbf{F}^{n+1} \mathbf{S}^{n+1}$ upon the solution at the new time step, this is actually a nonlinear equation. It is solved using a Newton iterative scheme based on a linearization of equation (38) with respect to velocities \mathbf{v}^{n+1} . The details of this procedure follow the standard approach for nonlinear problems and will not be presented here.

8.2 Time Integration for the Flow Domain

To advance the solution for the flow domain, a one-step implicit θ -method was used. It is based on an approximation of the rates of change of the solution as:

$$\dot{\mathbf{u}}^{n+1} = \frac{1}{\theta}(\mathbf{u}^{n+1} - \mathbf{u}^n) + (1 - \frac{1}{\theta})\dot{\mathbf{u}}^n \quad (38)$$

where $\theta \in (0, 1]$. It is a first order algorithm except for $\theta = \frac{1}{2}$, at which it has a second order time accuracy. Substitution of the above formula into the weak statement for the fluid leads to the following equation:

$$\begin{aligned} \left(\frac{\rho_F}{\theta} \mathbf{u}^{n+1}, \mathbf{w} \right) + c(\mathbf{u}^{n+1}, \mathbf{u}^{n+1}, \mathbf{w}) + a(\mathbf{u}^{n+1}, \mathbf{w}) + b(\mathbf{w}, p^{n+1}) &= \\ (f^{n+1}, \mathbf{w}) + \left(\frac{\rho_F}{\theta} \mathbf{u}^n - (\rho_F \dot{\mathbf{u}}^n, \mathbf{w}) + b.i. \right. & \\ \left. b(\mathbf{u}^{n+1}, q) \right) &= 0 \end{aligned} \quad (39)$$

where the consecutive terms were explained in Section 7.4.

The above equation calculates velocities \mathbf{u}^{n+1} at a new time step t^{n+1} . Due to the presence of nonlinear terms $c(\mathbf{u}^{n+1}, \mathbf{u}^{n+1}, \mathbf{w})$ this is actually a nonlinear equation. Similarly as for the solid subdomain, it is solved using a Newton iterative scheme based on linearization with respect to velocities \mathbf{v}^{n+1} .

8.3 Solution of Combined Flow-Structural Equations

A significant characteristic of the time-discretized structural and flow equations (38) and (40) is that they calculate velocities at the new time step t^{n+1} . Thus, the corresponding Newton equations can be discretized and solved for the entire domain simultaneously. In particular, discretization by the finite element shape functions leads to the following incremental matrix equations to be solved in each nonlinear iteration:

$$\begin{aligned} \mathbf{A}_i^{n+1} d\mathbf{U}_i^{n+1} &= \mathbf{R}_i^{n+1} \quad \text{in } \Omega, \\ \mathbf{U}_{i+1}^{n+1} &= \mathbf{U}_i^{n+1} + d\mathbf{U}_i^{n+1} \end{aligned} \quad (40)$$

where \mathbf{A} is the left-hand side matrix, \mathbf{U} is the discrete solution vector and \mathbf{R} is a nonlinear residual vector calculated according to the Newton procedure. The discrete solution vector \mathbf{U} represents velocities of both the fluid and solid subdomains. This simultaneous solution preserves the relevant two-way coupling between the fluid and the solid domain.

8.4 Moving Mesh Algorithm

Upon examination of equation (40) it can be seen that the grid or mesh velocity \mathbf{u}^g appears explicitly in this equation. During the solution of iterative equation (40), the grid velocity is treated as known. However, the distribution of mesh velocity and position need to be updated after each time step.

Importantly, for the sake of the solution of the nonlinear equation (40), the distribution of the mesh velocity is quite arbitrary. The only stringent requirement comes from the interface condition (34), which indicates that the grid velocity on the fluid-solid boundary is equal to the velocity of the solid.

In the solution process, the mesh motion is calculated after calculating the velocities \mathbf{u}^{n+1} . The typical mesh motion problem and boundary conditions are characterized by a Laplace equation for each of the components of mesh velocity:

$$u_{k,ii}^g = 0 \quad \text{in } \Omega^F \quad (41)$$

where $i, k = 1, N$ and u_k^g are the x,y,z components of the grid velocities. The boundary conditions are:

$$\begin{aligned} u_k^g &= v_k^S & \text{on } \Gamma^I \\ u_k^g &= 0 & \text{on } \Gamma^W \\ u_{k,n}^g &= 0 & \text{on } \Gamma^O \end{aligned}$$

Here Γ^I is the flow-structure interface, Γ^W is a fixed solid wall (if present) and Γ^O indicates fluid boundary with other boundary conditions, such as prescribed inflow, outflow or symmetry. The comma indicates a derivative and n denotes a direction normal to the boundary.

The above approach allows for modeling of multiple bodies with arbitrary surface/interface movements as one would expect in fluid-structure interaction problems.

8.5 Overall Solution Algorithm

The dynamic flow-structure interaction problem is solved via a time stepping algorithm, with a relevant update to the mesh to accomodate structural motion. The overall solution sequence is as follows:

1. Define a finite element mesh covering both the structural and flow domains in some initial configuration. Identify the material properties and boundary conditions.
2. Define an initial flow solution by solving the steady-state flow equations or importing a flow solution from an independent flow solver.

3. Perform time stepping from time $t = 0$ to the final time $t = T$ using time increments dt . For each time step:
 - (a) calculate the velocities for the entire domain by solving the flow-structure interaction problem (40),
 - (b) based on the calculated velocities, determine new structural positions \mathbf{r} ,
 - (c) update the structural reference configuration Ω to correspond to the new structural positions, and
 - (d) calculate the mesh velocity \mathbf{u}^g from equation (41) and update the mesh location.

The above sequence can be augmented by an additional perturbation of the steady-state equilibrium configuration (to trigger instabilities), saving the solution to a file or performing postprocessing. Moreover, at selected time intervals the mesh adaptation can be performed.

9 Underwater Shock Analysis

9.1 Introduction

The development of numerical techniques to assess damage to naval structures due to underwater explosions is a continuing effort to ensure that our naval forces are placed at less risk in combat. Underwater shock damage typically combines the effects of the incident shock wave that propagates at high velocity outward from the explosive source and envelops the structure with later low frequency pulsations of the explosion gas bubble. The shock wave induces high frequency response in the structure with possible localized damage while the bubble oscillation excites low frequency beam modes of the entire vessel that can lead to bend-buckling phenomena. Another source of damage is due to the possible collapse of the gas bubble on the structure resulting in fluid impact. This latter effect can be the most severe result of an underwater explosion if it occurs very close to the structure.

Naval vessels are typically constructed using discretely stiffened plate and shell modules. As a result, the effect of an underwater shock is to induce deformations in the structure that, if they are large enough, will produce compressive loads in the stiffeners causing the torsional-flexural instability called "tripping." One result of this process is the tendency of the stiffener to tear away from the hull plating that may or may not open the hull to flooding. In such a scenario the study of the combined effects of underwater shock and crack propagation is of vital interest to the naval community. The interfacing of two computer codes that address these phenomena separately is therefore appropriate to study the interaction of these combined effects. These codes are the Underwater Shock Analysis (USA) code developed and supported by Unique Software Applications of Colorado Springs, Colorado, and the PHLEXcrack code

developed and supported by the Computational Mechanics Company of Austin, Texas. For more details on the theory and application of USA code, one can refer to [10].

9.2 Underwater Shock Analysis Code

The USA code determines the transient response of a totally or partially submerged structure due to an incident acoustic shock wave and subsequent explosion gas bubble pulsation. USA is built around the family of Doubly Asymptotic Approximations (DAA) that treat the fluid structure interaction by a boundary element approach that avoids the incorporation of a large volume of fluid around the structure and instead, works through the interchange of wet-surface variables only that are passed back and forth between the DAA equation solver and that of the structural analyzer.

The DAA approach to underwater shock analysis involves the exact fluid-structure interaction relations in the limit of zero and infinite frequency while making a smooth transition between these limits in the mid-frequency range. The most typically used members of the DAA family available to the user are the one first order DAA1, and two variants of a second order more accurate DAA2. These latter are a so-called "modal" form DAA2m, and a "curvature corrected" form DAA2c. Both the DAA1 and the DAA2m use symmetric matrices in their formulations while the DAA2c requires a nonsymmetric solver package. In terms of increasing accuracy, computation time, and memory requirements these choices rank in the order DAA1, DAA2m, and DAA2c, respectively. For special purposes, the limiting forms of the DAA1, the high frequency Plane Wave Approximation (PWA), and the low frequency Virtual Mass Approximation (VMA) are also available to the user.

A preprocessor of the USA code, FLUMAS, constructs a fluid mass matrix that fully couples all of the fluid node points on the wet surface of the structure to describe the low frequency asymptote of the DAA through Newton's law of motion. This matrix is symmetric and positive definite and is produced by a boundary element formulation of Laplace's equation for the irrotational flow generated in an inviscid, incompressible fluid by motions of the structure. The fluid mass matrix also includes the primary effects of element curvature that enhance the accuracy of the low frequency response.

The USA code incorporates a variety of symmetry and boundary conditions in its wet-surface fluid model. Half and quarter models are easily treated to save computational resources for symmetric problems. In addition, free surface effects are included through an imaging process. The code also has a bottom effects model but this can only be regarded as a crude approximation to reality in the absence of any real information.

9.3 Structural Response Equation

The discretized differential equation system for the dynamic response of a linear structure can be expressed in the form [10]

$$\mathbf{M}_s \ddot{\mathbf{x}} + \mathbf{C}_s \dot{\mathbf{x}} + \mathbf{K}_s \mathbf{x} = \mathbf{f}$$

where \mathbf{x} is a structural displacement vector, \mathbf{M}_s , \mathbf{C} and \mathbf{K}_s are the symmetric linear structural mass, damping and stiffness matrices, respectively. \mathbf{f} is the external force vector, and a (\cdot) denotes temporal derivative.

For excitation of a submerged structure by an acoustic wave, \mathbf{f} is given by

$$\mathbf{f} = -\mathbf{G}\mathbf{A}_f(\mathbf{p}_I + \mathbf{p}_s) + \mathbf{f}_D$$

where \mathbf{p}_I and \mathbf{p}_s are nodal pressure vectors for the wet-surface fluid mesh pertaining to the (known) incident wave and the (unknown) scattered wave, respectively, \mathbf{f}_D is the dry structure applied force vector, \mathbf{A}_f is the diagonal area matrix associated with elements in the fluid mesh, and \mathbf{G} is the transformation matrix that relates the structural and fluid surface forces.

The Doubly Asymptotic Approximation may be written [50,51]

$$\mathbf{M}_f \dot{\mathbf{p}}_s + \rho c \mathbf{A}_f \mathbf{p}_s = \rho c \mathbf{M}_f \dot{\mathbf{u}}_s$$

where \mathbf{u}_s is the vector of scattered-wave fluid particle velocities normal to the structure's wet surface, ρ and c are the density and sound velocity of the fluid, respectively, and \mathbf{M}_f is the symmetric fluid mass matrix for the wet-surface fluid mesh.

For excitation by an incident acoustic wave, \mathbf{u}_s is related to structural response by the kinematic compatibility relation

$$\mathbf{G}^T \dot{\mathbf{x}} = \mathbf{u}_I + \mathbf{u}_s$$

Generally, \mathbf{G} is a rectangular matrix whose height greatly exceeds its width, inasmuch as the number of structural DOF usually considerably exceeds the number of fluid DOF. \mathbf{G} is constructed such that only the translational DOF for the structure couple with the fluid DOF.

Combining the above equations yields the interaction equations

$$\mathbf{M}_s \ddot{\mathbf{x}} + \mathbf{C}_s \dot{\mathbf{x}} + \mathbf{K}_s \mathbf{x} = -\mathbf{G} \mathbf{A}_f (\mathbf{p}_I + \mathbf{p}_s)$$

$$\mathbf{M}_f \dot{\mathbf{p}}_s + \rho c \mathbf{A}_f \mathbf{p}_s = \rho c \mathbf{M}_f (\mathbf{G}^T \ddot{\mathbf{x}} - \mathbf{u}_I)$$

These equations may be solved simultaneously at each step, but such a procedure is exceedingly expensive, because of the large connectivity of the coefficient matrix involved. Efficient computation is possible through the application of a staggered solution procedure, that is, unconditionally stable with respect to the choice of time increment, at least for linear structural response. Staggered solution procedure is effected from the above equations (assuming that \mathbf{M}_s is nonsingular) by extracting $\mathbf{G}^T \ddot{\mathbf{x}}$ from the first equation into the latter one. Premultiplication of the resulting equation by $\mathbf{A}_f \mathbf{M}_f^{-1}$ then yields

$$\mathbf{A}_f \dot{\mathbf{p}}_s + (\mathbf{D}_{f1} + \mathbf{D}_s) \mathbf{p}_s = -\rho c \mathbf{A}_f \mathbf{G}^T \mathbf{M}_s^{-1} (\mathbf{C}_s \dot{\mathbf{x}} + \mathbf{K}_s \mathbf{x}) - \mathbf{D}_s \mathbf{p}_I - \rho c \mathbf{A}_f \dot{\mathbf{u}}_I$$

where

$$\mathbf{D}_{f1} = \rho c \mathbf{A}_f \mathbf{M}_f^{-1} \mathbf{A}_f$$

$$\mathbf{D}_s = \rho c \mathbf{A}_f \mathbf{G}^T \mathbf{M}_s^{-1} \mathbf{G} \mathbf{A}_f$$

Note that both \mathbf{D}_{f1} and \mathbf{D}_s are symmetric. This process of injecting one of the coupled equations into the other to achieve stability is termed *augmentation*, hence the above is called are herein termed *augmented interaction equations*.

The solution of the above equation is carried in the USA routines, while PHLEX code provides time integration of the elastic structure (with the crack) and provides to the USA module the value of the vector $(\mathbf{C}_s \dot{\mathbf{x}} + \mathbf{K}_s \mathbf{x})$ with possible dry structure force \mathbf{f}_D . USA module in turn returns the vector of wet surface pressures $(\mathbf{p}_I + \mathbf{p}_s)$

9.4 USA-PHLEXcrack Interface

At this time the interface between USA and PHLEXcrack incorporates only the capability to treat structural models that are constructed with quad faces on the wet-surface boundary. Triangular faces are treated as a subset of the quad data structures. The Surface-Of-Revolution

(SOR) simplification for structural beam models has not been implemented nor has the capability to treat cavitation in the surrounding fluid. Coupling of a general DAA boundary to a beam model of a structure for whipping analysis has also been left out of the current interface. This initial effort to couple USA with PHLEXcrack has concentrated on only the most basic type of analysis to demonstrate the utility of the combined software package. The above features of the USA code can be added to the interface package as the need arises.

The interface allows for different time steps to be used in USA and PHLEXcrack with all necessary interpolations carried out on the USA side of the interface. Typically the PHLEXcrack time step may be smaller than the USA step size.

Special class of boundary conditions, called "WetFace," has been created in PHLEXcrack for this purpose. Those BC's should be defined on element faces defining the surface of the body that will be subjected to the shock wave. The amount of pressure applied on these surfaces are determined interactively from USA code.

PHLEXcrack is dimensionally neutral, i.e. it accepts quantities and mesh dimensions in any unit system, as long as all of them are defined in the consistent way. USA code, on the other hand, being CFD code, requires that the quantities are defined in dimensionless, normalized units. In the examples presented in the manual we assumed the characteristic size of the model (outer radius of the shell), fluid density, and shock wave speed in the fluid as all equal 1.0. Thus the unit of time is half of the amount of time it takes for the shock wave to travel in fluid from top to the bottom of the cylinder, the density of the shell material is 7.85, etc.

9.5 Computational Procedure

Usage of the combined USA-PHLEXcrack software involves the following multi-step procedure:

- 1.) Preparation of a structural model
- 2.) Run PHLEXcrack in a pre-processing mode to create a database file that is accessed by two USA preprocessors, FLUMAS and AUGMAT
- 3.) Run FLUMAS to create a fluid mass matrix appropriate to the wet-surface geometry of the structural model as well as a variety of subsidiary fluid data
- 4.) Run AUGMAT to create the so-called "augmented matrices" that are used in the computational form of the DAA
- 5.) Restart the USA-PHLEXcrack processor to conduct the underwater shock analysis with incident wave data and explosive charge location supplied by USA input records
- 6.) Post-process transient response results using PHLEXcrack utilities and the USA post-processor POSTPR

Input records for all USA processors are listed in a series of manuals that describe each variable and the order in which they must appear in the input files. The POSTPR utility incorporates the capability to compute pseudo-velocity shock spectra that is of interest to those who design equipment mounts and is not generally found in a typical off-the-shelf post-processors.

9.6 Software Validation

Two standard benchmark problems have been studied to test the new interface. These are the infinite cylindrical shell and the spherical shell, both excited by a plane step wave. Closed form modal solutions for both problems have been discussed in the literature and the USA code has demonstrated its capabilities by replicating those solutions with a variety of linear and nonlinear structural codes in the past. Computations with the USA-PHLEXcrack software exhibit the same excellent comparisons with these closed form solutions and other computational simulations with only the typical differences caused by a variety of element formulations used by each structural code.

10 Mixed Elements in PHLEXcrack: EPM and Classical Shell Elements

PHLEXcrack can handle both EPM three dimensional elements and the shell elements combined together in the same model. Being meshless, EPM elements are naturally suited for modeling crack propagation. The advantages are that cracks can advance in any direction and are not confined to element faces. While shell elements are reserved to model areas of thin plates. Both static and dynamic analyses can be conducted on the combined model.

11 User Routines for Crack Propagation

The computer program delivered for this project is setup in such a way to allow the user to add his or her own routines for crack propagation. The user can add up to two routines for the calculation of fracture quantities (SIF's) and up to two routines for propagation physics.

The outline of these routines has been coded and is delivered along with the code. In addition, the source code for routines of the fracture methods and physics described in Sections 5 and 6 are delivered. They are intended to be used for guidance in writing routines for similar purposes.

Once a new routine is written, it has to be compiled and linked with the code. The

user can then use the new routine by setting some parameters in the tcl file. Please refer to the User Manual for information on these parameters.

12 Controlling the accuracy of numerical algorithms

Based on limited set of numerical experiments (some of them are presented in section 13) we present here some practical notes on the limitations of the code, and on proper selection of various control parameters. In particular element sizes, although significantly larger than in traditional FEM models, should be selected appropriately to local crack accuracy.

12.1 Crack details and element sizes

Although EPM allows to place crack arbitrarily inside the element, and it calculates proper element stiffness matrices (or their equivalents in EPM), it cannot properly account for all minute details of crack shape within element. Our experiments suggest that any details of crack surface (sudden turns, sharp corners, etc.) which are smaller than approximately 1/10 of the element size are neglected in the solution. Crack curvature should not be larger than comparable to element size, as well as crack having large variations of curvatures within single element, will be not solved with very high accuracy.

12.2 Crack shape and accuracy of SIF estimation

As explained in section 5, stress intensity factors are computed on the basis of averaged stress in the vicinity of crack front, and are based on the assumption that the crack within this vicinity will be closely approximated by the planar, two-dimensional problem. The size of the domain for the SIF calculation (see parameters description in "PHLEXcrack manual") should be adjusted appropriately to local curvatures of the crack surface. The length of the cylindrical domain is adjusted automatically (to the length of the straight segment of crack front), but the diameter of the cylinder and other parameters have to be selected by the user.

12.3 Crack smoothness

In order to improve stability of crack growth, and limit the restrictions presented in the last section, PHLEXcrack contains several algorithms ensuring proper smoothness of the crack:

- the angle between to consecutive crack velocity vectors is limited,

- velocities between neighboring points along crack edge are smoothed out using weighted averaging between neighbors,
- lengths of the segments along the crack front are adjusted to fit between user prescribed bounds, and also based on the local curvature of the crack front.

To see results of automatic “refinement” of the crack front, see bottom and right parts of the crack presented in the Fig. 54

12.4 Crack folds

Crack propagation algorithm is similar to the explicit time integration method (next position of the crack front is calculated based on the stresses computed at current crack position). Fortunately, time stepping required for the dynamic fracture is usually sufficiently small, and the problem may arise in pseudo-static analysis (i.e. without accurate modeling of shock waves). There is however another similar restriction, resulting from the discrete (segmented) approximation of the crack front. In general, user should adjust parameters to ensure that crack movement within one time step (velocity times delta time) is smaller than the minimal crack segment length. Fig. 54, top portion, shows exactly this case, when crack propagation vectors were too long compared with segment length and crack “folded” on the small scale (comparable with the segment length). PHLEXcrack geometry engine is not designed to handle such geometries, and this type of results will most probably cause internal code error and crash.

Crack engine can handle properly intersection of the crack with the boundary, or with other parts of the crack. When such intersection is detected, the crack front is stopped at the node, split into smaller multiple fronts if necessary, and solution can proceed. User should however verify, that the above described accuracy and smoothness requirements are satisfied, otherwise further computations may have no physical meaning.

13 Numerical Examples

The cracked element partition method (CEPM) described previously is used in this section to solve several illustrative examples. In all examples, the stress intensity factors are computed using the technique outlined in Section 5.

13.1 Single Crack with Mode I Solution Under Static Loading

In this section, the edge-cracked panel illustrated in Fig. 11 is analyzed using the CEPMP. In the computations, the following parameters are assumed: $h = b = 1.0$, $a = 0.5$, distributed tractions $\sigma = 1.0$ and uniform thickness $t = 0.1$. The material is assumed to be linearly elastic with $E = 1000.0$ and $\nu = 0.3$.

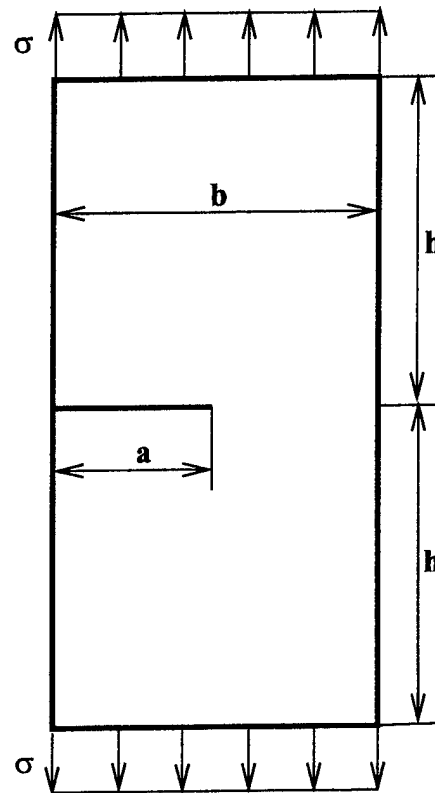


Figure 11: Single edge-notch test specimen

The domain is discretized using the hexahedral mesh shown in Fig. 12. There are 961 elements in the mesh. A state of plane strain is modeled by constraining the displacement in the z -direction at $z = 0$ and $z = t$. The representation of the crack surface is shown in Fig. 13. It is composed of four triangles and two edge elements (used to identify the crack front). Note that these triangles are used only for the geometric representation of the crack surface and that there is no degrees of freedom associated with them. The stress intensity factors reported for this problem are computed at $\mathbf{x} = (0.5, 1.0, 0.05)$ which are the coordinates of a node of an edge element and is located at the middle plane of the body. The geometric definition of the outer skin of the domain (shown on Fig. 13) is also given as input data for the geometric engine. The base vectors of the coordinate system associated with singular functions used at the crack front is also displayed in Fig. 13. The base vectors and corresponding singular functions are

computed completely automatically using the geometric engine. This functionality is specially important during dynamic crack propagation simulations or when the geometry of the domain or crack surface are not so trivial as in this example.

Figure 14 shows a closer look at the discretization near the crack front. It can be observed that the crack surface does not respect the element boundaries—*It can arbitrarily cut the elements in the mesh.* The nodes carrying singular degrees of freedom are represented by diamond-shaped dots. The singular functions used at these nodes are those presented at Section 3. Whenever they are used, only the nodes of the elements that contain the crack front are enriched with these singular shape functions (in this example, there is only one element at the crack front). The enrichment of the elements at the crack front with appropriate singular functions is done automatically using the geometric engine to construct appropriate coordinate systems at the crack front.

The computed values of K_I and of the strain energy, U , for several discretization are shown on Tables 1 and 2. In the tables, N denotes the number of degree of freedom of a particular discretization. The discontinuity in the displacement field for those elements completely severed by the crack surface is modeled using the technique described in Section 2.4. For the element at the crack front, two approaches are used. In the first case, the visibility approach in combination or not with singular functions at the crack front is used. The results using this approach is shown in Table 1. In the second case, the wrap-around approach in combination with singular functions at the crack front is used. The results using this approach is shown in Table 2. The notation $\mathbf{p} = 1 + p_x, 1 + p_y, 1 + p_z = \mathbf{1} + (p_x, p_y, p_z)$ is used to denote the polynomial order of the approximation over *non-cracked* elements. The “1’s” indicating the linear order of the functions defining the partition of unity (linear hexahedral finite element shape functions in this case) and p_x, p_y, p_z denote the degrees of the polynomial basis functions $L_{i\alpha}$ in the x, y, z directions, respectively (see Section 2.3 for the definition of $L_{i\alpha}$). Therefore, if only the partition of unity is used, we have $\mathbf{p} = 1 + 0, 1 + 0, 1 + 0 = \mathbf{1} + \mathbf{0}$. A quadratic approximation in the plane XY and linear in the z -direction is denoted by $\mathbf{p} = 1 + 1, 1 + 1, 1 + 0 = \mathbf{1} + (1, 1, 0)$. For the cracked elements, the polynomial degree of the approximation is denoted by $\mathbf{p}_c = 0 + p_x, 0 + p_y, 0 + p_z = \mathbf{0} + (p_x, p_y, p_z)$. The “0’s” indicating that, in general, the functions defining the partition of unity over cracked elements can only represent a constant function. A quadratic approximation in the plane XY and linear in the z -direction over cracked elements (fully or partially cracked) is denoted by $\mathbf{p}_c = 0 + 2, 0 + 2, 0 + 1 = \mathbf{0} + (2, 2, 1)$. The stress intensity factors are computed using the technique outlined in Section 5. The following parameters are used in all computations presented in this section:

- Dimensions of the extraction domain: $\mathbf{d} = (0.4, 1.0, 0.05)$. Which represents a cylinder of radius 0.4 and length 0.05.
- Number of integration points in the r, θ and z directions: $\mathbf{n} = (10, 20, 1)$, respectively.
- Type of \mathbf{D} matrix equal identity matrix

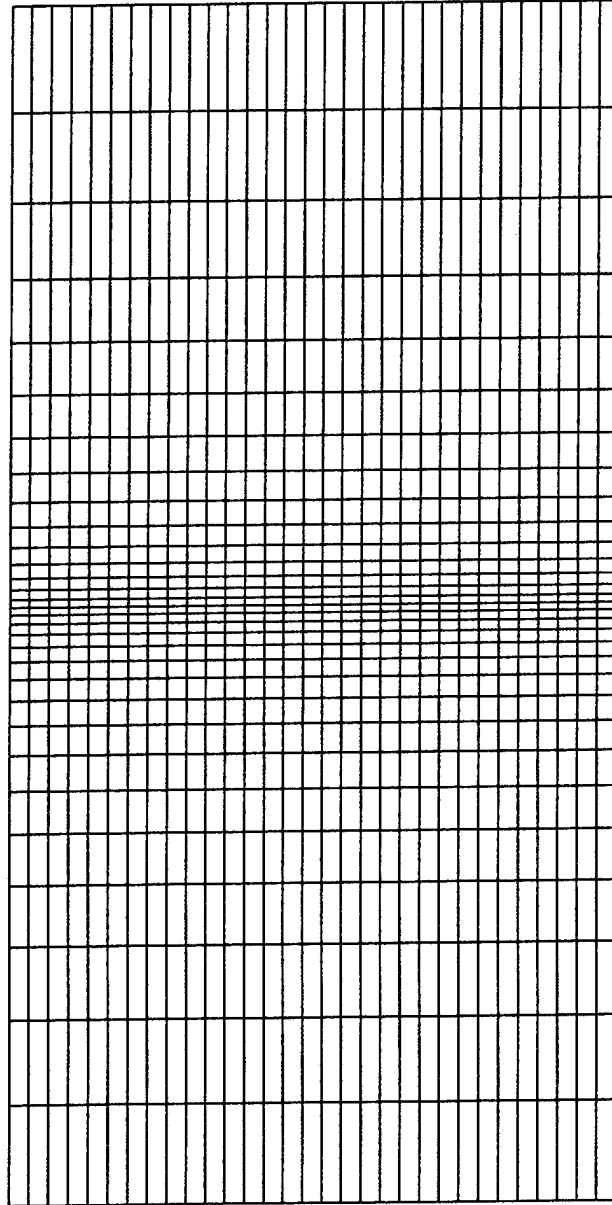


Figure 12: CEPM discretization for the model problem shown on Fig. 11.

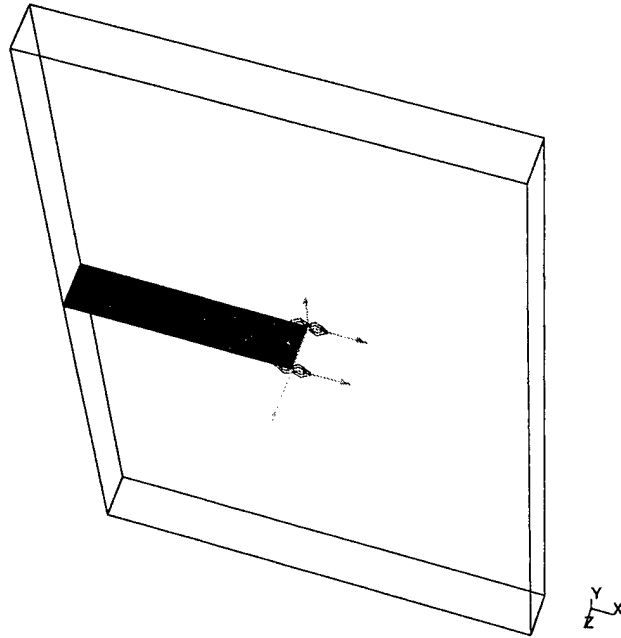


Figure 13: Triangularization of the crack surface.

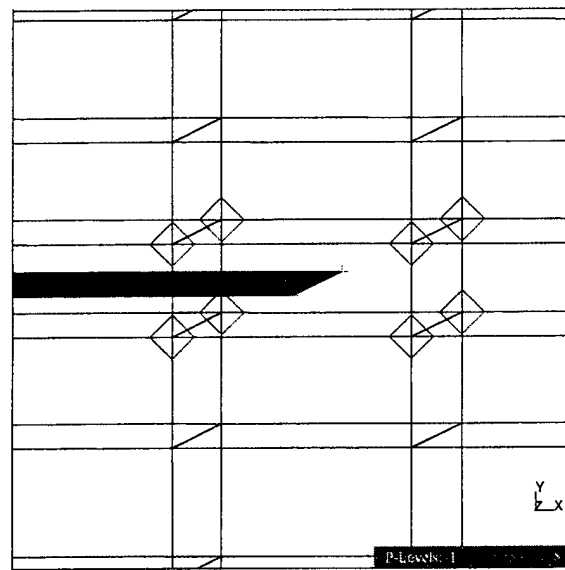


Figure 14: Zoom at the crack front.

- Power of the weighting function: 6.

As a reference, the value of K_I computed by Tada et al. [48] using a boundary technique applied to a two dimensional idealization of the problem shown on Fig. 11. The value $K_I^{Tada} = 3.54259$ reported in [48] has an error smaller than 0.5%.

The discretizations Vis-1, Vis-2 and Vis-3 do not use singular functions at the crack front while the discretizations Vis-2, Vis-4 and Vis-6 do. All 'Vis' discretizations use the visibility approach to model the crack at partially cut elements. It can be observed from Table 1 that the use of singular functions gives a noticeable improvement on the computed stress intensity factors while the increase in the number of degrees of freedom is only marginal (less than one percent in the case of the discretization Vis-5). In contrast, the p-enrichment of the cracked elements only (discretizations Vis-1 and Vis-3) gives little improvement on the computed K_I . Nonetheless, the enrichment does improve the computed strain energy by about 3%. This behavior indicates that the technique used to compute the stress intensity factor is not optimal since optimally the computed stress intensity factors must converge at the same rate as the computed strain energy [46, 47].

The discretizations WA-1, WA-2 and WA-3 use singular functions in combination with the wrap-around technique to model the crack at partially cut elements. Comparing the results for the discretizations Vis-2 with WA-1 or Vis-4 with WA-2, it can be observed that, for the same number of degrees of freedom, the wrap-around approach gives better results for the stress intensity factors than the visibility approach. This is in spite of the fact that the discretizations using wrap-around has a smaller strain energy than the corresponding discretizations with using visibility. This can be explained by the fact that the visibility approach creates spurious discontinuities in the displacement field near the crack front which results in a less stiff discretization compared with the wrap-around approach (which does not create such spurious discontinuities). It can be observed that the discretization WA-2 gives a better value for K_I than the discretization WA-3 in spite of the fact that the latter gives a larger value for the strain energy than the latter. This, again, points to limitations of the technique used to compute the stress intensity factor.

Figure 15 shows a contour plot of the displacement in the vertical direction near the crack computed using the discretization WA-3. The discontinuity in the displacement field constructed using the technique presented in Section 2.4 is clearly observed. Figure 16 shows a contour plot for the von Mises stress computed with the same discretization and Fig. 17 shows a closer look near the crack front. The computed stresses are all raw stresses computed at arbitrary points inside each element. Figure 18 shows the same quantity computed using the discretization vis-5. It can be observed that the stress field is quite disturbed near the crack front. This is caused by the spurious discontinuities created by the visibility approach near the crack front.

Table 1: CEPMP using visibility for the elements at the crack front. The stress intensity factor is computed at (0.5, 1.0, 0.05).

Discret.	$\mathbf{p} = \mathbf{1}+$	$\mathbf{p}_c = \mathbf{0}+$	Sing Fn	N	$U \times 10^4$	K_I	K_I/K_I^{Tada}
Vis-1	(0,0,0)	(1,1,1)	No	6,756	2.27136	3.2387	0.91422
Vis-2	(0,0,0)	(1,1,1)	Yes	6,900	2.32941	3.3741	0.95244
Vis-3	(0,0,0)	(2,2,1)	No	7,776	2.33848	3.2495	0.91727
Vis-4	(0,0,0)	(2,2,1)	Yes	7,920	2.36701	3.3938	0.95800
Vis-5	(1,1,0)	(2,2,1)	No	19,656	2.38643	3.4339	0.96932
Vis-6	(1,1,0)	(2,2,1)	Yes	19,800	2.42281	3.4584	0.97623

Table 2: CEPMP using wrap-around for the elements at the crack front. The stress intensity factor is computed at (0.5, 1.0, 0.05).

Discret.	$\mathbf{p} = \mathbf{1}+$	$\mathbf{p}_c = \mathbf{0}+$	Sing Fn	N	$U \times 10^4$	K_I	K_I/K_I^{Tada}
WA-1	(0,0,0)	(1,1,1)	Yes	6,900	2.22465	3.3886	0.95653
WA-2	(0,0,0)	(2,2,1)	Yes	7,920	2.25596	3.4228	0.96618
WA-3	(1,1,0)	(2,2,1)	Yes	19,800	2.29396	3.3431	0.94369

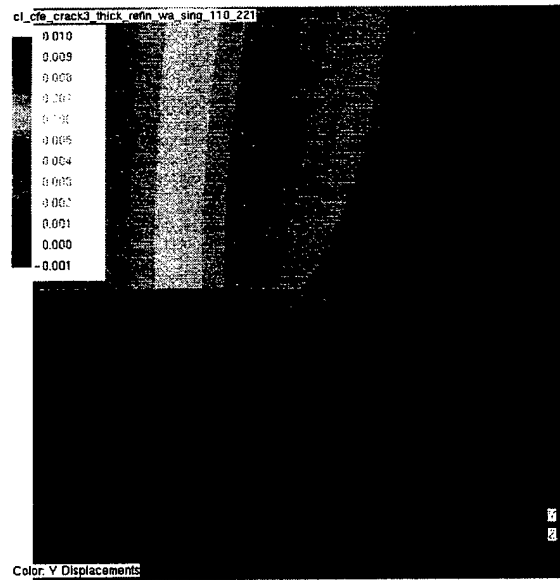


Figure 15: Displacement in the y -direction computed with discretization WA-3.

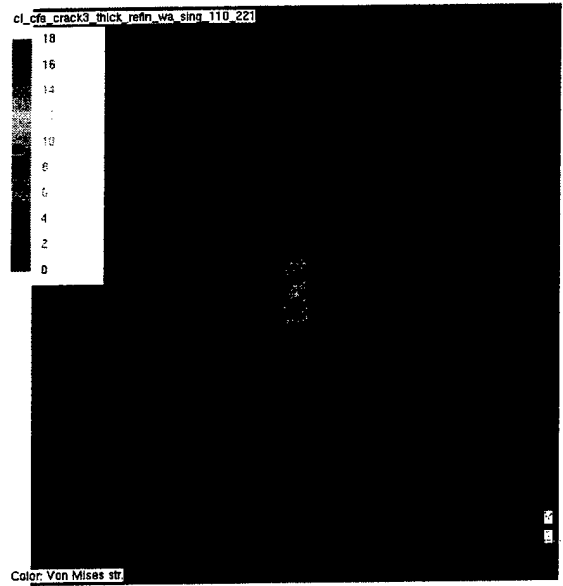


Figure 16: Von Mises stress computed with discretization WA-3.

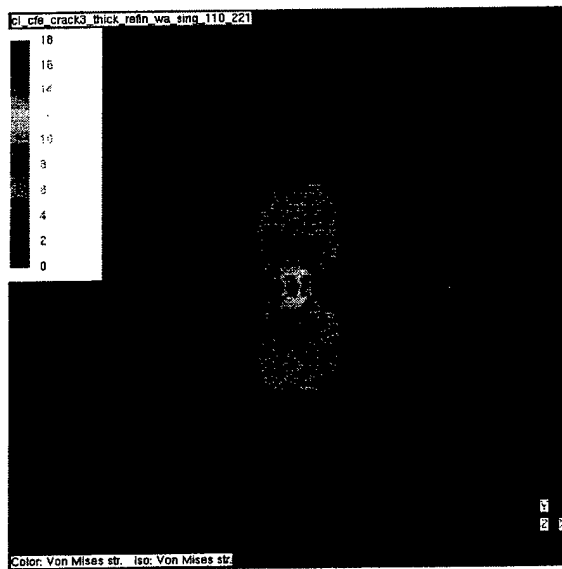


Figure 17: Zoom at the crack front showing von Mises stress computed with discretization WA-3.

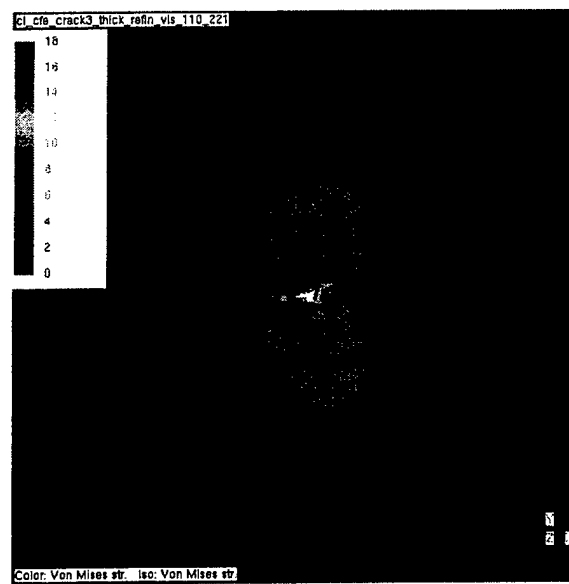


Figure 18: Zoom at the crack front showing von Mises stress computed with discretization vis-5.

13.2 An Inclined Crack Problem

As another test problem, we consider the cracked panel shown in Fig. 19. This problem has been analyzed by Szabó and Babuška [46] using the p -version of the finite element method and by Oden and Duarte [37] using the hp Cloud method. In both references, plane stress condition and unity thickness are used. Here, the plane stress condition is approximated by using a small thickness, $t = 0.1$, for the domain compared to other dimensions. In addition, we assume Young's modulus $E = 1$, Poisson's ratio $\nu = 0.3$, distributed traction $\sigma = 1.0$ and $w = 1$ (see Fig. 19). These same values are used in references [46] and [37]. We adopt as a reference, the values of K_I , K_{II} and strain energy, U , computed by Oden and Duarte [37]. They are, respectively,

$$\begin{aligned} K_I^{Ref} &= 1.508284 \\ K_{II}^{Ref} &= -0.729706 \\ U^{Ref} &= 0.170402 \end{aligned}$$

These values agree very well with those computed by Szabó and Babuška [46] (less than 0.1% difference). The value of the U^{Ref} was scaled to take into account the difference in thickness used here ($t = 0.1$) and adopted by Oden and Duarte [37] ($t = 1.0$).

The discretization of the domain using 765 hexahedral elements is shown on Fig. 20. The inclined crack is also shown in the figure. Figure 21 shows a closer look near the crack. It can be observed that the crack surface cuts the elements in the mesh in a quite arbitrary manner. In fact, the meshing of the domain is done as if there is no crack at all. The only consideration used during the meshing of the domain was to use a more refined mesh near the location of the crack front. The crack representation is created and passed to the geometric engine as input data. The geometric engine uses no information whatsoever about the mesh. Nodes in the mesh that are too close to the crack surface are then automatically moved a small distance away from the crack surface (this can be observed in Fig. 21 near the crack front). This is required because an approximation node must be located at one or another side of the crack surface (but not (numerically) on it). The representation of the crack surface used here is topologically identical to the one used in the previous example (see Fig. 13). Figure 21 shows a closer look at the mesh and crack representation near the crack front. The nodes carrying singular degrees of freedom are represented by diamond-shaped dots. The singular functions used at these nodes are those presented at Section 3. As in the previous example, whenever they are used, only the nodes of the elements that contain the crack front are enriched with these singular shape functions.

Figure 23 shows a contour plot of the displacement in the vertical direction near the crack computed using the discretization WA-3. The discontinuity in the displacement field constructed using the technique presented in Section 2.4 is clearly observed. Figure 24 shows a contour plot for the von Mises stress computed with the same discretization and Fig. 25 shows a closer look near the crack front. The computed stresses are all raw stresses computed at arbitrary points inside each element. Figure 26 shows the same quantity computed using

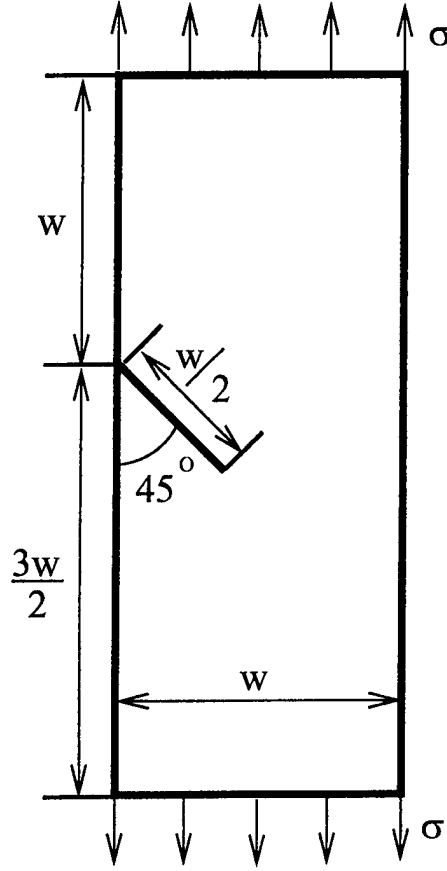


Figure 19: Problem definition.

the discretization vis-5. It can be observed, as in the previous example, that the stress field is quite disturbed near the crack front. This is caused by the spurious discontinuities created by the visibility approach near the crack front.

A summary of the results is presented on Tables 3, 4, 5 and 6. The notation used to describe the various discretizations used is the same as in the previous section. The stress intensity factors are computed at a point in the crack front located at the middle surface of the body. The following parameters are used for extracting the stress intensity factors:

- Dimensions of the extraction domain: $\mathbf{d} = (0.2, 1.0, 0.05)$. Which represents a cylinder of radius 0.4 and length 0.05.
- Number of integration points in the r, θ and z directions: $\mathbf{n} = (10, 40, 1)$, respectively.
- Type of \mathbf{D} matrix equal identity matrix
- Power of the weighting function: 6.

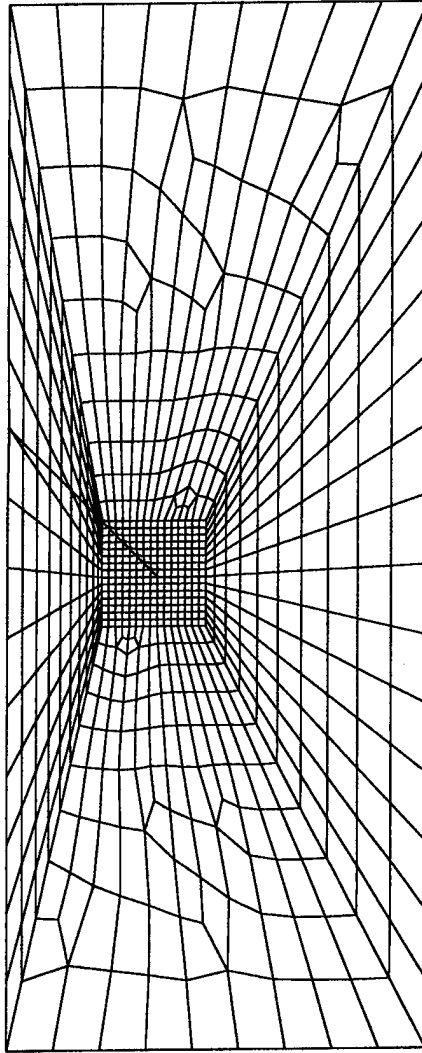


Figure 20: CEPM discretization for the model problem shown on Fig. 19.

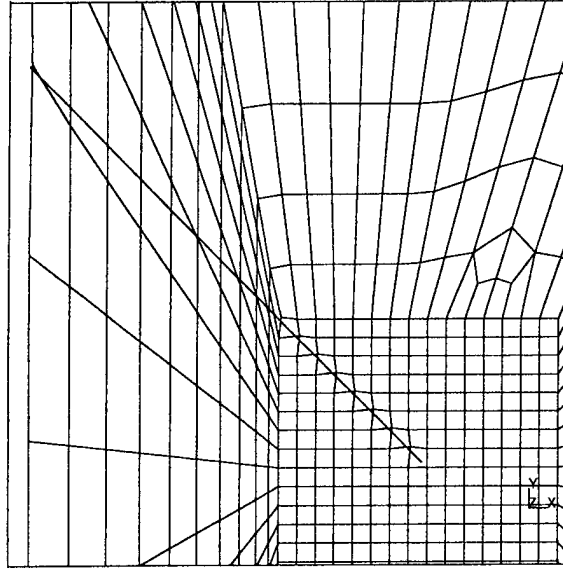


Figure 21: Zoom showing the elements cut by the crack surface.

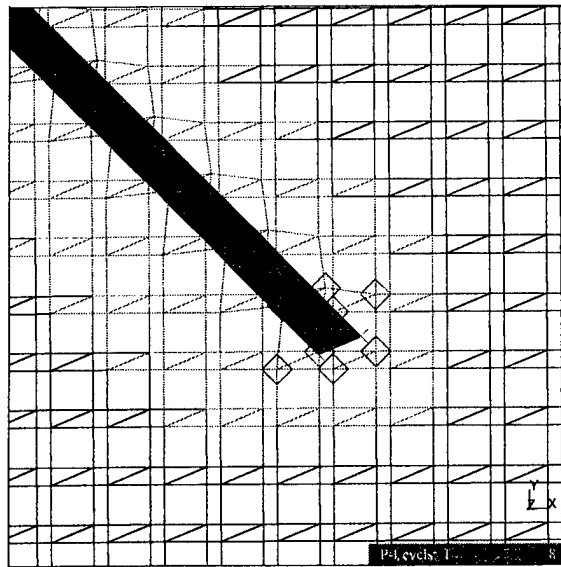


Figure 22: Three-dimensional view of the crack front.

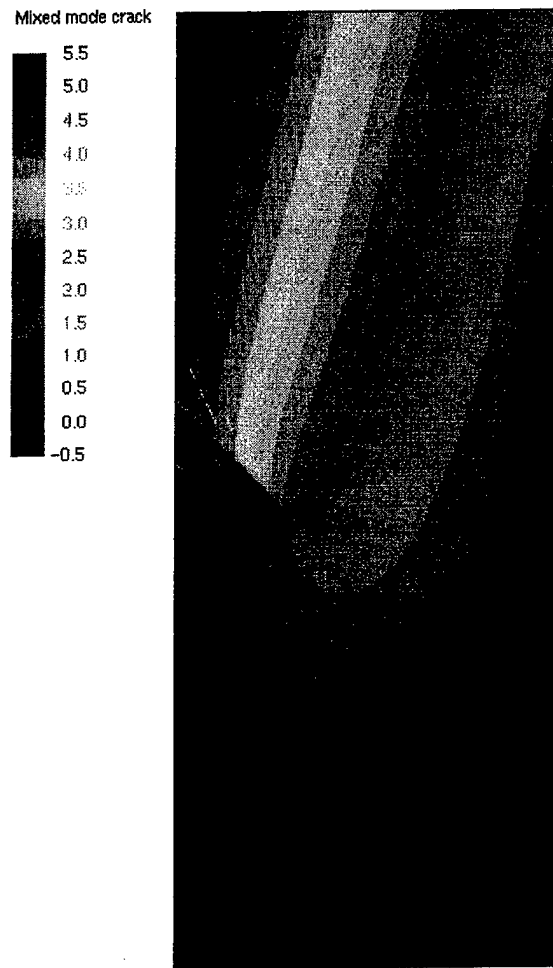


Figure 23: Contour plot of the displacement in the vertical direction computed using the discretization WA-3.

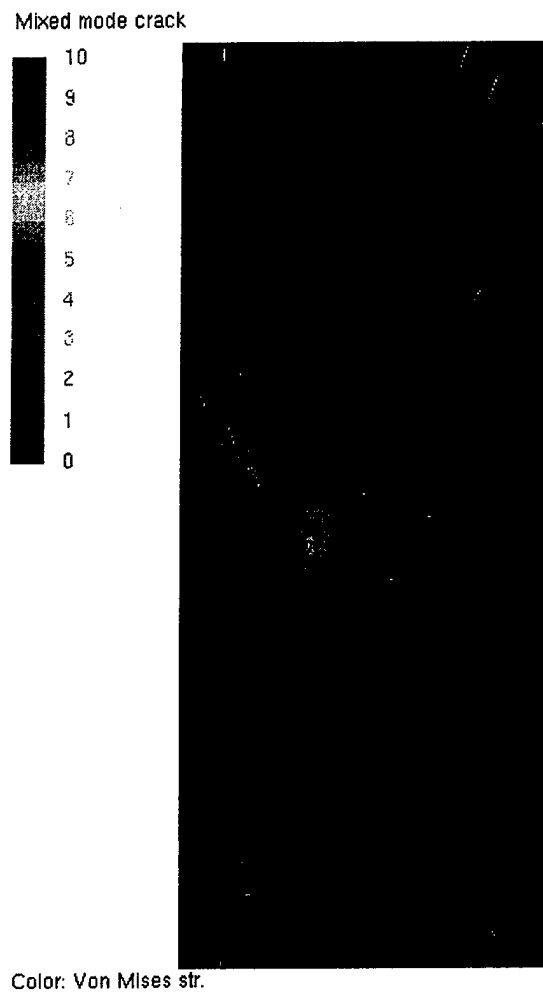


Figure 24: Contour plot for the von Vises stress field computed with discretization WA-3.

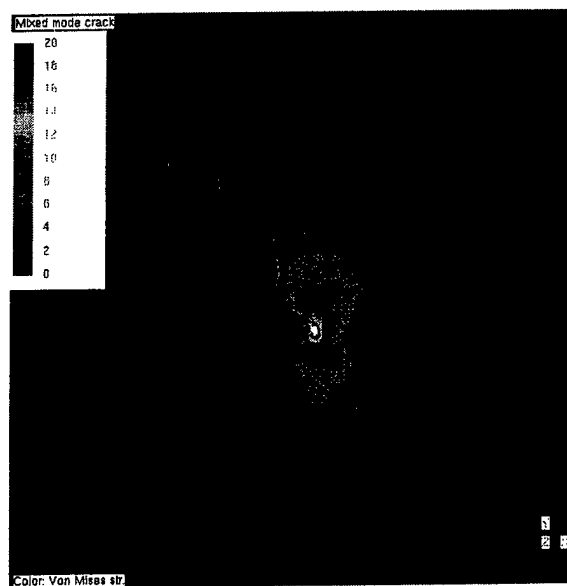


Figure 25: Zoom showing the von Mises stress near the crack front computed with discretization WA-3.

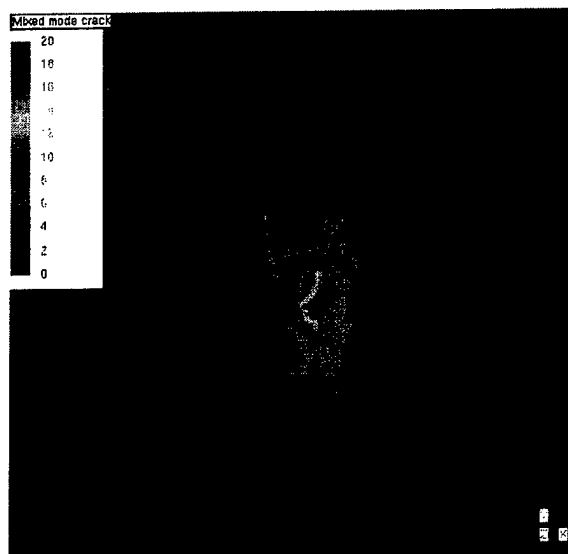


Figure 26: Zoom showing the von Mises stress near the crack front computed with discretization vis-5.

Table 3 shows the computed strain energy for the various discretizations using the visibility approach at the crack front with or without singular functions. It can be observed that the p -enrichment of the approximation has a more significant effect on the strain energy values than the addition of singular functions at the crack front. Nonetheless, as in the previous example, the addition of singular functions improves considerably the computed stress intensity factors (Cf. Table 4). In the case of the discretization Vis-3, for example, the enrichment with singular functions adds only 1.8% more degrees of freedom while the error on the compute value of K_I decreases from 14.0% to only 3.5% and the error on the computed K_{II} decreases from 10.8% to only 1.4%. That is, the error on the computed K_I and K_{II} decrease by 75.0% and 87.0%, respectively.

The results for the discretizations that use the wrap-around approach at the crack front are presented in Tables 5 and 6. The discretization WA-3 has a relative error in energy, $(U^{Ref} - U)/U^{Ref}$, of only 0.02% which corresponds to a relative error in the energy norm of only 1.41%. This same problem was also solved using the classical hp finite element method with the hp adaptation driven by error indicators based on the element residual method. The results obtained after seven adaptive cycles are shown on Table 7. The relative error in energy and in the energy norm for this discretization are 0.3% and 5.5%, respectively. Note that this discretization has more degrees of freedom (18585) than the discretization WA-3 (16632) but an error in the energy norm almost four times bigger. The reason for this is that the CEPD discretization can capture the singular field near the crack front more efficiently by using customized singular functions.

Table 3: CEPD using visibility for the elements at the crack front. In the table, “Discr” stands for discretization, “Sig Fn” stands for singular functions at the crack front, “N” stands for number of equations and “U” stands for strain energy.

Discr	$\mathbf{p} = 1+$	$\mathbf{p}_c = 0+$	Sig Fn	N	U	U/U^{Ref}
Vis-1	(0,0,0)	(1,1,1)	No	5874	0.157986	0.9271
Vis-2	(0,0,0)	(1,1,1)	Yes	6018	0.158644	0.9310
Vis-3	(0,0,0)	(2,2,1)	No	7764	0.164454	0.9651
Vis-4	(0,0,0)	(2,2,1)	Yes	7908	0.165053	0.9686
Vis-5	(1,1,0)	(2,2,1)	No	16488	0.171148	1.0044
Vis-6	(1,1,0)	(2,2,1)	Yes	16632	0.171558	1.0068

Table 4: CEPM using visibility for the elements at the crack front. In the table, "Discr" stands for discretization, "Sig Fn" stands for singular functions at the crack front and "N" stands for number of equations.

Discr	$\mathbf{p} = \mathbf{1}+$	$\mathbf{p}_c = \mathbf{0}+$	Sig Fn	N	K_I	K_{II}	K_I/K_I^{Ref}	K_{II}/K_{II}^{Ref}
Vis-1	(0,0,0)	(1,1,1)	No	5874	1.1099	-0.58755	0.7286	0.8052
Vis-2	(0,0,0)	(1,1,1)	Yes	6018	1.2656	-0.68936	0.8391	0.9447
Vis-3	(0,0,0)	(2,2,1)	No	7764	1.2964	-0.65095	0.8595	0.8921
Vis-4	(0,0,0)	(2,2,1)	Yes	7908	1.4550	-0.71979	0.9647	0.9864
Vis-5	(1,1,0)	(2,2,1)	No	16488	1.4904	-0.73524	0.9881	1.0076
Vis-6	(1,1,0)	(2,2,1)	Yes	16632	1.6595	-0.77266	1.1002	1.0589

Table 5: CEPM using wrap-around for the elements at the crack front. In the table, "Discr" stands for discretization, "Sig Fn" stands for singular functions at the crack front, "N" stands for number of equations and "U" stands for strain energy.

Discr	$\mathbf{p} = \mathbf{1}+$	$\mathbf{p}_c = \mathbf{0}+$	Sig Fn	N	U	U/U^{Ref}
WA-1	(0,0,0)	(1,1,1)	Yes	6018	0.157978	0.9271
WA-2	(0,0,0)	(2,2,1)	Yes	7908	0.164183	0.9635
WA-3	(1,1,0)	(2,2,1)	Yes	16632	0.170372	0.9998

Table 6: CEPM using wrap-around for the elements at the crack front. In the table, "Discr" stands for discretization, "Sig Fn" stands for singular functions at the crack front and "N" stands for number of equations.

Discr	$\mathbf{p} = \mathbf{1}+$	$\mathbf{p}_c = \mathbf{0}+$	Sig Fn	N	K_I	K_{II}	K_I/K_I^{Ref}	K_{II}/K_{II}^{Ref}
WA-1	(0,0,0)	(1,1,1)	Yes	6018	1.3175	-0.64084	0.8735	0.8782
WA-2	(0,0,0)	(2,2,1)	Yes	7908	1.4663	-0.69115	0.9722	0.9472
WA-3	(1,1,0)	(2,2,1)	Yes	16632	1.6290	-0.72814	1.0800	0.9979

Table 7: Results using hp finite element method and seven adaptive cycles.

Discr	N	U	U/U^{Ref}	K_I	K_{II}	K_I/K_I^{Ref}	K_{II}/K_{II}^{Ref}
hp FEM	18585	0.16988	0.9969	1.4680	-0.7041	0.9733	0.9649

13.3 Plate Under Impact Load

In this section we investigate the performance of the CEPm in modeling propagating cracks in a body subjected to impact loads. The test problem is illustrated on Fig. 27. This problem has been analyzed by Lu et al. [32], Krysl and Belytschko [24], Organ [44] and Belytschko and Tabbara [5] using the element free Galerkin method, by Gallego and Dominguez [21] using a boundary element method, among others. A state of plane strain and the following parameters are adopted

- Dimensions: $b = 10.0$, $h = 2.0$, $a = 5.0$ and uniform thickness $t = 0.1$,
- Loading: $\sigma(t) = \hat{\sigma} H(t) = 63750.0 H(t)$, $t \geq 0$. Here, $H(t)$ is the Heaviside step function.
- Material Properties: Linear elastic material with $E = 2.0 \times 10^{11}$, $\nu = 0.3$ and $\rho = 7833.0$.
- Time Step: $\Delta t = 10^{-5}$.

A state of plane strain is modeled by constraining the displacement in the z -direction at $z = 0$ and $z = t$. Two uniform hexahedral meshes are used. The first one has 125, 49 and 1 element in the x -, y - and z -directions, respectively. This same mesh was used in the computations of Krysl and Belytschko [24]. We denote this as the *fine mesh*. The second mesh has 65, 25 and 1 element in the x -, y - and z -directions, respectively and is shown on Fig. 28 along with the crack representation. This mesh is denoted as the *coarse mesh*. The representation of the crack surface and of the outer skin of the body are shown in Figs. 29 and 30. It is composed of five triangles and four edge elements. There are five vertex nodes uniformly spaced at the crack front. The stress intensity factors reported for this problem are computed at $\mathbf{x} = (5.0, 2.0, 0.05)$ and is located at the middle plane of the body. This vertex node is denoted *vertex 3* hereafter.

The stress intensity factors are computed using the technique outlined in Section 5. The following parameters are used in all computations presented in this section:

- Dimensions of the extraction domain: $\mathbf{d} = (1.0, 1.0, 0.05)$. Which represents a cylinder of radius 1.0 and length 0.05.
- Number of integration points in the r, θ and z directions: $\mathbf{n} = (5, 30, 1)$, respectively.
- Type of \mathbf{D} matrix equal material matrix
- Power of the weighting function: 3.

Four CEPm discretizations are used (the notation used for \mathbf{p} and \mathbf{p}_c is defined in Section 13.1) :

- Discretization 1:
 - Fine mesh ($125 \times 49 \times 1$ elements),
 - Degree of approximation over non cracked elements $\mathbf{p} = \mathbf{1} + \mathbf{0}$.
 - Degree of approximation over cracked elements $\mathbf{p}_c = \mathbf{0} + \mathbf{1}$.
 - Algorithm to build a discontinuous partition of unity over the elements at the crack front: Visibility (and no singular functions).
- Discretization 2:
 - Fine mesh ($125 \times 49 \times 1$ elements),
 - Degree of approximation over non cracked elements $\mathbf{p} = \mathbf{1} + \mathbf{0}$.
 - Degree of approximation over cracked elements $\mathbf{p}_c = \mathbf{0} + \mathbf{1}$.
 - Algorithm to build a discontinuous partition of unity over the elements at the crack front: Wrap-around and singular functions.
- Discretization 3:
 - Coarse mesh ($65 \times 25 \times 1$ elements),
 - Degree of approximation over non cracked elements $\mathbf{p} = \mathbf{1} + \mathbf{0}$.
 - Degree of approximation over cracked elements $\mathbf{p}_c = \mathbf{0} + \mathbf{1}$.
 - Algorithm to build a discontinuous partition of unity over the elements at the crack front: Visibility (and no singular functions).
- Discretization 4:
 - Coarse mesh ($65 \times 25 \times 1$ elements),
 - Degree of approximation over non cracked elements $\mathbf{p} = \mathbf{1} + \mathbf{0}$.
 - Degree of approximation over cracked elements $\mathbf{p}_c = \mathbf{0} + (2, 2, 1)$.
 - Algorithm to build a discontinuous partition of unity over the elements at the crack front: Visibility (and no singular functions).

13.3.1 Reference Solution

The CEPD results are compared to the analytic solution for a semi-infinite crack in the plane proposed by Freund [20]. The problem solved by Freund is represented in Fig. 31. The two-dimensional domain has a straight semi-infinite crack, is in a state of plane strain and is

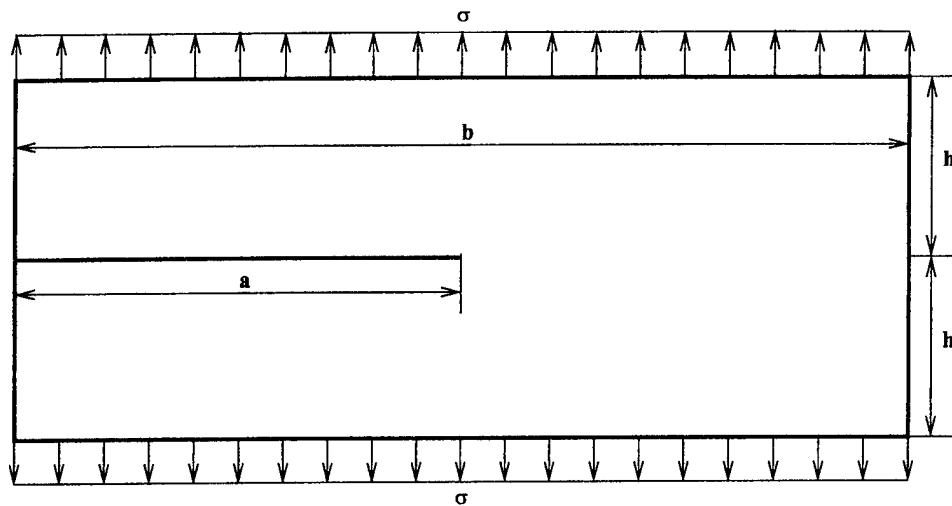


Figure 27: Model problem used to analyze the performance of the CEPM in modeling propagating cracks in a body subjected to impact loads.

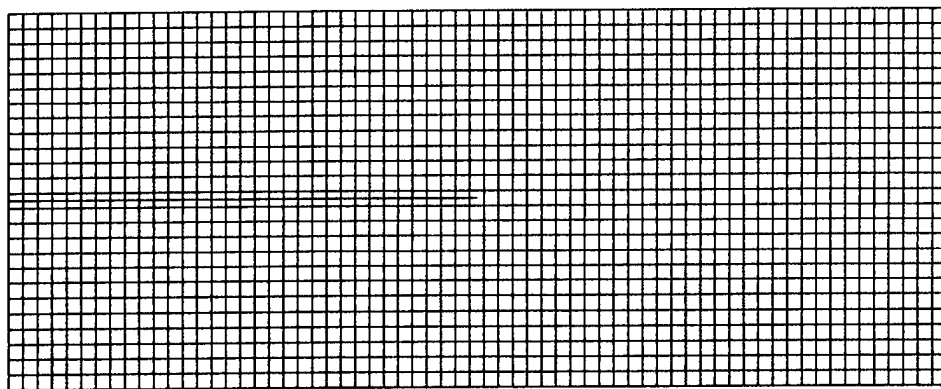


Figure 28: CEPM Discretization.

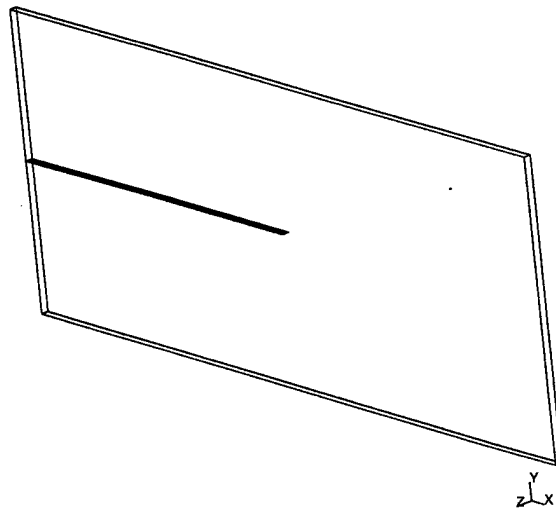


Figure 29: Crack Representation.

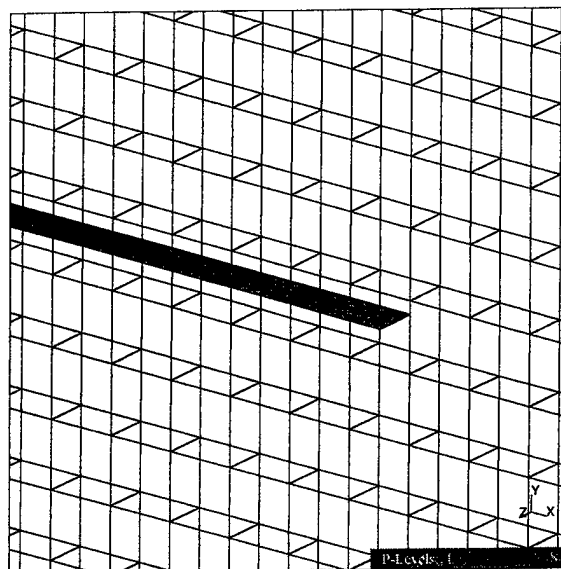


Figure 30: CEPM Discretization.

loaded by uniformly distributed tractions applied at time $t = 0$. The mode-I stress intensity factor, as a function of time and crack speed C is given by ([20]).

$$K_I(t, C) = \frac{4\hat{\sigma}H(t - \hat{t})k(C)}{1 - \nu} \sqrt{\frac{(1 - 2\nu)(t - \hat{t})C_d}{\pi}}$$

where,

- $\hat{\sigma}$ is the magnitude of the tensile tractions,
- C_d is the pressure wave speed in the body which is given by

$$C_d = \sqrt{\frac{\mu(\kappa + 1)}{\rho(\kappa - 1)}}$$

where $\mu = \frac{E}{2(1+\nu)}$ and $\kappa = 3 - 4\nu$ (for plane strain state). For the material properties given previously, we get $C_d = 5862.7$.

- \hat{t} is the time the elastic wave hits the crack. For the problem represented in Fig. 27 and $C_d = 5862.7$

$$\hat{t} = 0.000341$$

- $k(C)$ is scaling factor that takes into account that the crack front is advancing with speed C and is given by

$$k(C) = \frac{1.0 - C/C_R}{1.0 - 0.5 * C/C_R}$$

where C_R is the Rayleigh wave speed. For this test problem, with the material properties given above, $C_R = 3030$ (see Section 6).

Due to symmetries, $K_{II} = 0$. The magnitude of the energy release rate is given by

$$G(C, t) = \left(1 - \frac{C}{C_R}\right) G(C = 0, t)$$

where

$$G(C = 0, t) = \frac{K_I^2(C = 0, t)}{E^*}$$

and, for plane strain,

$$E^* = \frac{E}{1 - \nu^2}$$

It should be noted that due to the finite dimensions of the domain modeled here there will be waves reflected by the boundary. This reflected waves will eventually reach the crack front and a comparison of the numerical solution with the above reference solution will no

longer be valid. The first reflected wave to reach the crack is a pressure wave after traveling from a loaded edge to the opposite edge and then back to the crack front [24]. This happens at

$$\bar{t} = \frac{3h}{C_d} = \frac{6}{5862.7} = 0.00102$$

A more detailed discussion on the wave patterns that reach the crack front can be found in [24].

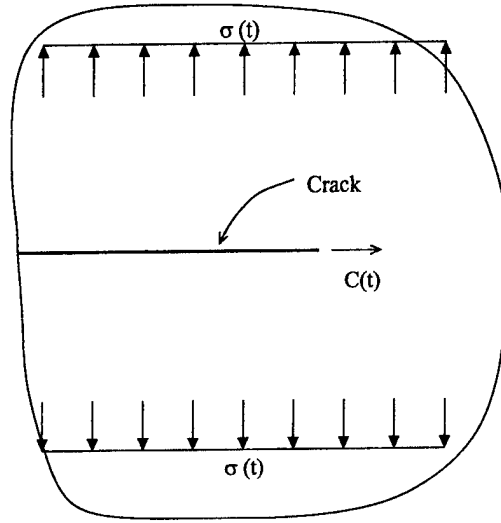


Figure 31: Definition of reference problem.

13.3.2 Stationary Crack

As a first test, we consider the case of a stationary crack. Discretizations 1 and 2 as described above are used. The computed mode-I and mode-II stress intensity factors and the energy release rate G are plotted on Figs. 32 and 33, respectively. It can be observed a good agreement between the computed and reference values for $t < \bar{t}$. The finite dimensions of the extraction domain for the stress intensity factors and of the support of the shape functions are responsible for the non-zero values computed before the pressure wave hits the crack front (at $t = \bar{t}$). It can also be observed that both discretizations give very close results.

13.3.3 Moving Crack with Prescribed Speed and Direction

Here, the crack speed is given by

$$C(t) = H(t - 0.00044) \frac{C_R}{3} = H(t - 0.00044) 1010.0$$

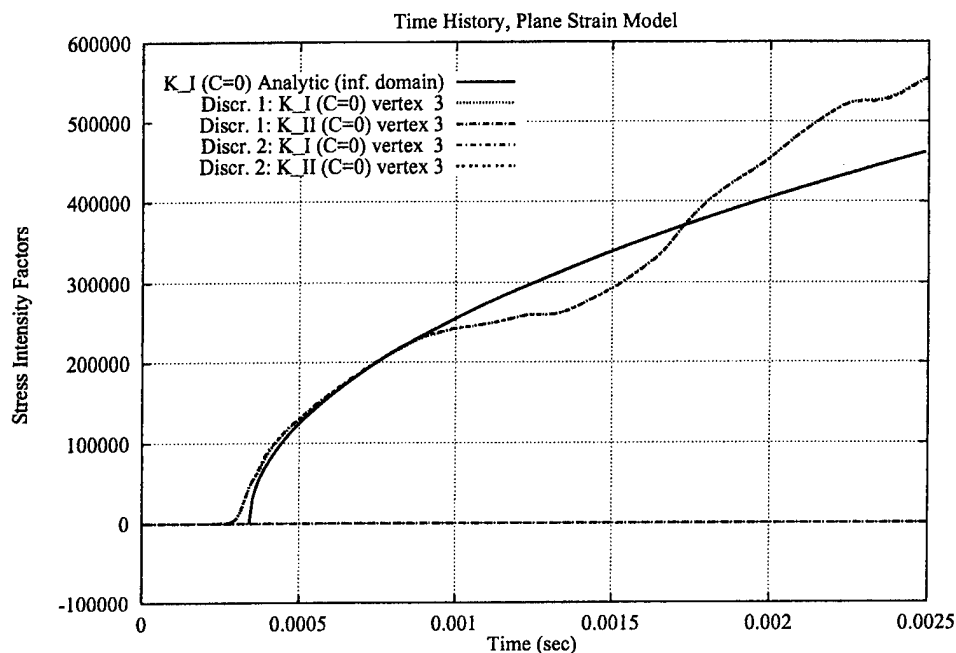


Figure 32: Time history for mode-I and mode-II stress intensity factors K_I and K_{II} using Discretizations 1 and 2. Stationary crack.

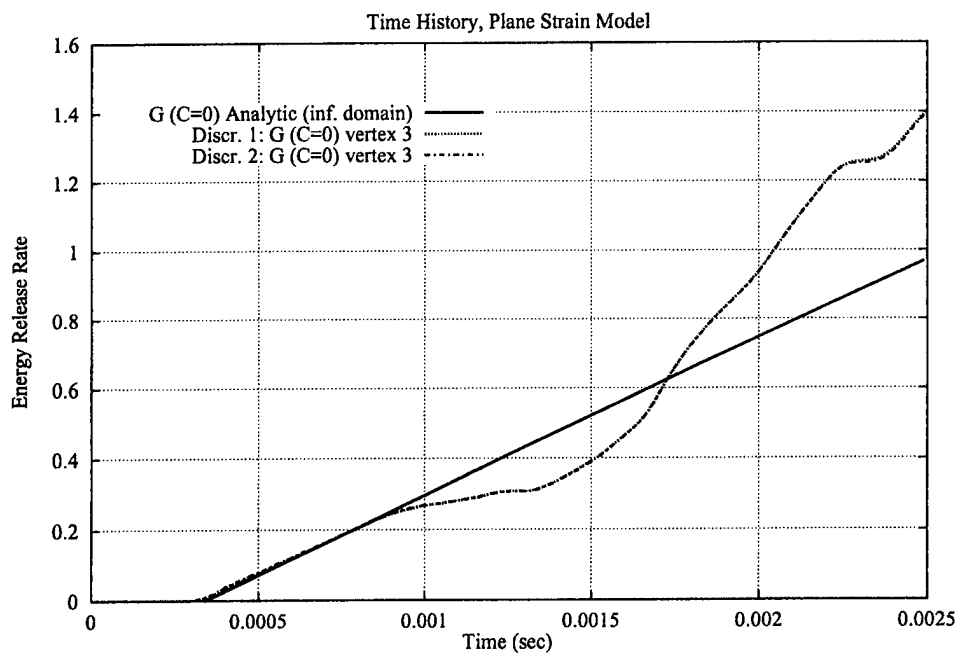


Figure 33: Time history for the energy release rate G using Discretizations 1 and 2. Stationary Crack.

and propagates straightly along the direction $\theta = 0$ where θ is measured from the x -axis counter-clockwise. The values of K_I , K_{II} and G computed with the Discretization 1 are plotted in Figs. 34 and 35 against the reference solution. While the computed values present some oscillation, they are in good agreement with the reference solution before $t = \hat{t}$ (when reflected waves hits the crack surface). Organ [44], Krysl and Belytschko [24] and Belytschko and Tabbara [5] also reported oscillations on their results obtained with the element free Galerkin method.

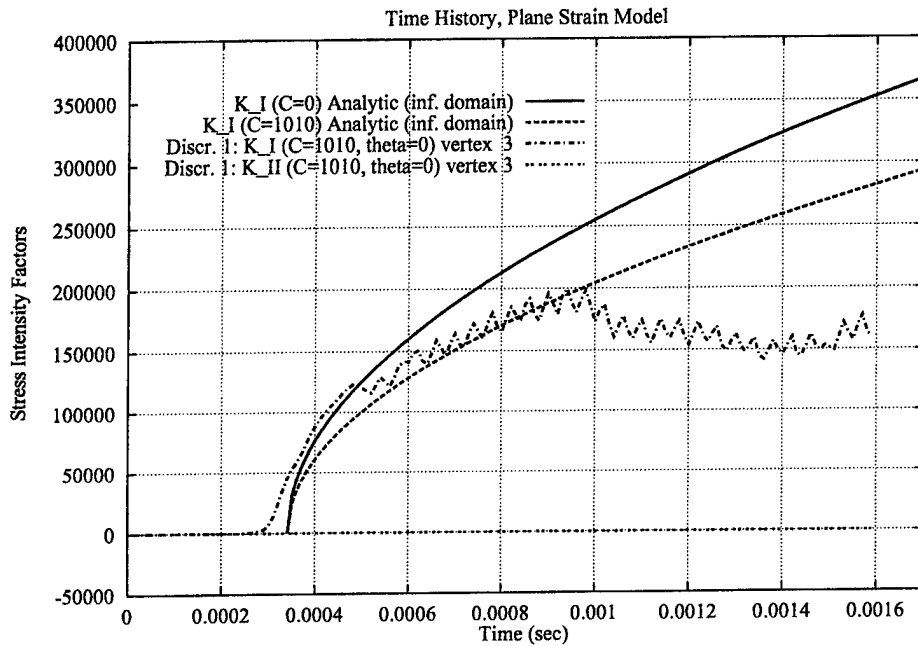


Figure 34: Time history for mode-I and mode-II stress intensity factors K_I and K_{II} using Discretization 1. Moving Crack with prescribed speed and direction.

13.3.4 Moving Crack with Prescribed Speed

Here, the direction of the crack advancement is given by (10). Discretization 1 (fine mesh) and 3 (coarse mesh) are used. Figures 36 and 37 shows the time history for K_I , K_{II} and G , respectively. It can be observed that CEPM solution was able to capture very well the slope of the reference solution. Figure 38 shows the crack surface at time $t = 0.00168s$.

The results obtained with Discretization 3 (coarse mesh) are shown on Figs. 39 and 40. The approximation of the slope of the reference solution is still quite reasonable, specially while the crack remains stationary ($t < 0.00044$). The results obtained with Discretizations 1 and 3 are compared on Figs. 41 and 42.

The effect of p enrichment of the cracked elements is investigated by using Discretization

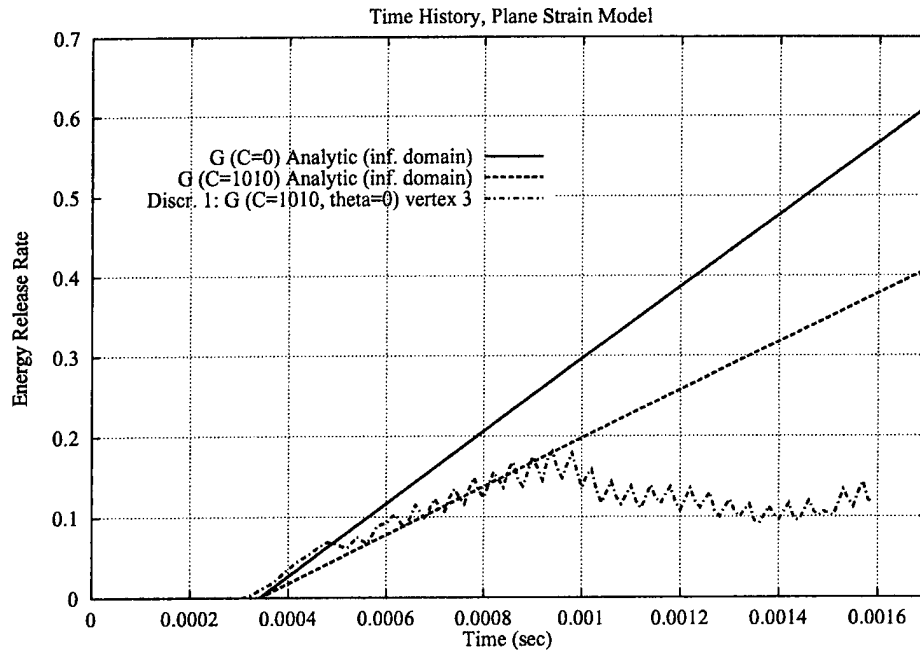


Figure 35: Time history for the energy release rate G using Discretization 1. Moving Crack with prescribed speed and direction.

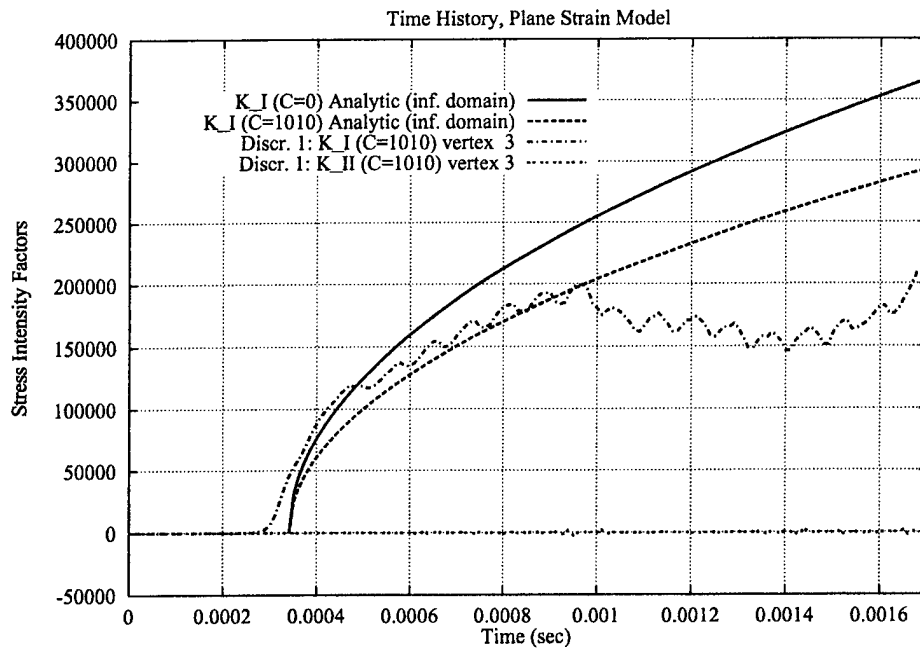


Figure 36: Time history for K_I and K_{II} using Discretization 1. Moving crack with advancement direction given by (10).

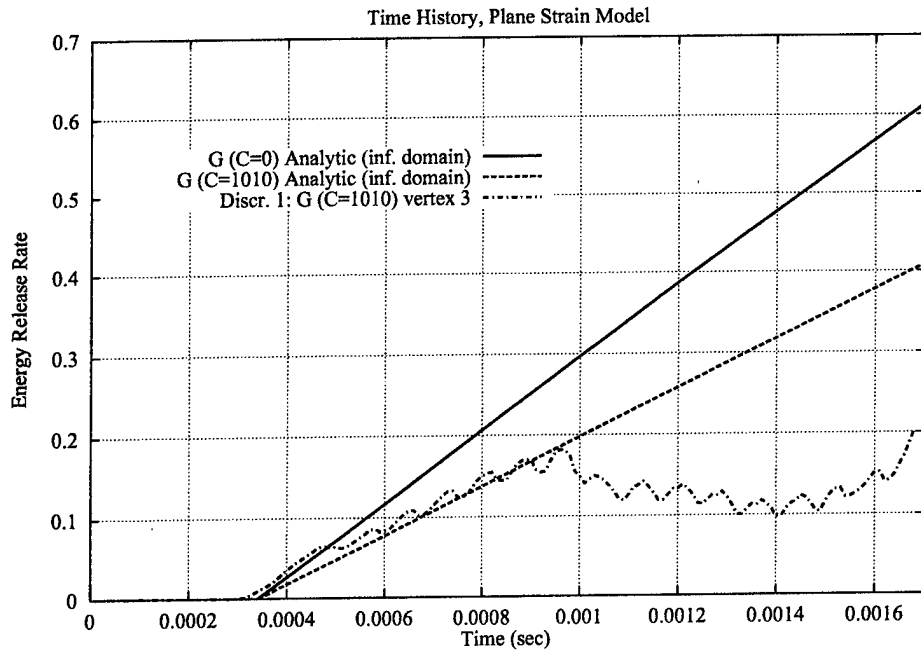


Figure 37: Time history for G using Discretization 1. Moving crack with advancement direction given by (10).

4 (which is identical to Discretization 3 but uses $\mathbf{p}_c = \mathbf{0} + (2, 2, 1)$ instead of $\mathbf{p}_c = \mathbf{0} + \mathbf{1}$. The results for this discretization are shown on Figs. 43 and 44. It can be observed the the p enrichment of the cracked elements increases the amplitude of the oscillations of the computed quantities. Figure 45 is identical to Fig. 44 but here the tic marks of the x-axis are placed exactly at the times when the crack front crossed the boundary between two elements. It can be observed that the peaks in the oscillations occur just before those instants. Note that as the crack crosses the boundary between two elements it passes close to the nodes. This same phenomena was observed by Organ [44].

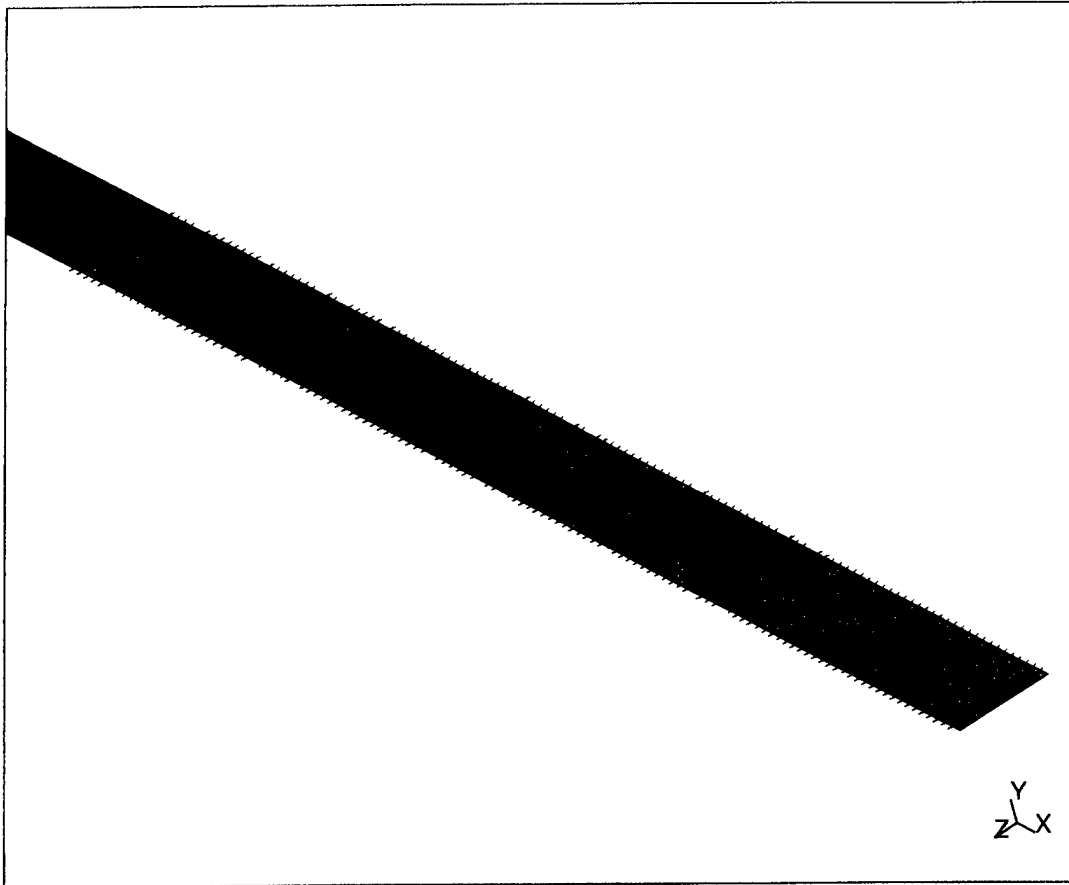


Figure 38: Crack surface at $t = 0.00168s$. Moving crack with advancement direction given by (10).

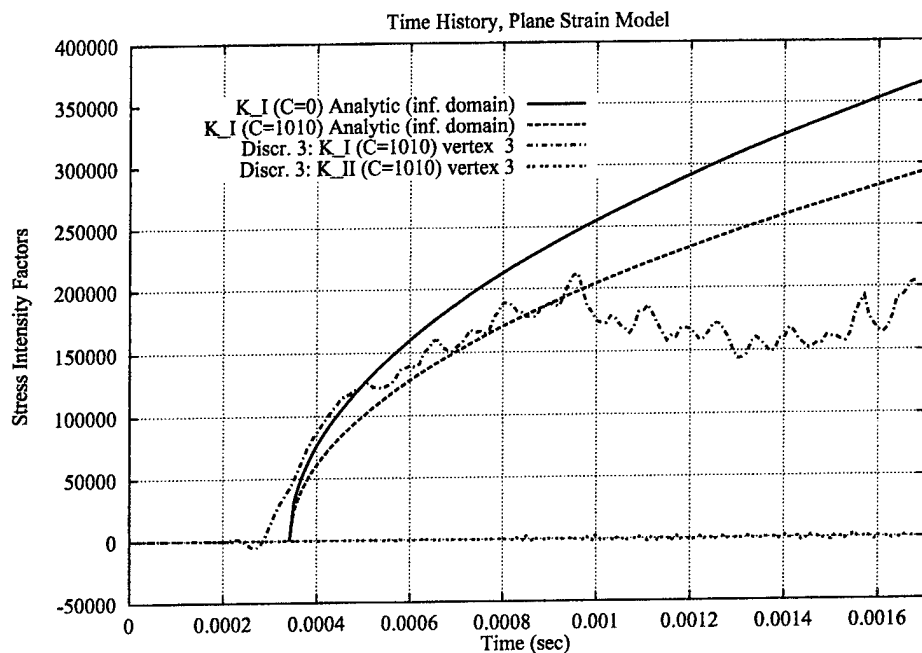


Figure 39: Time history for K_I and K_{II} using Discretization 3. Moving crack with advancement direction given by (10).

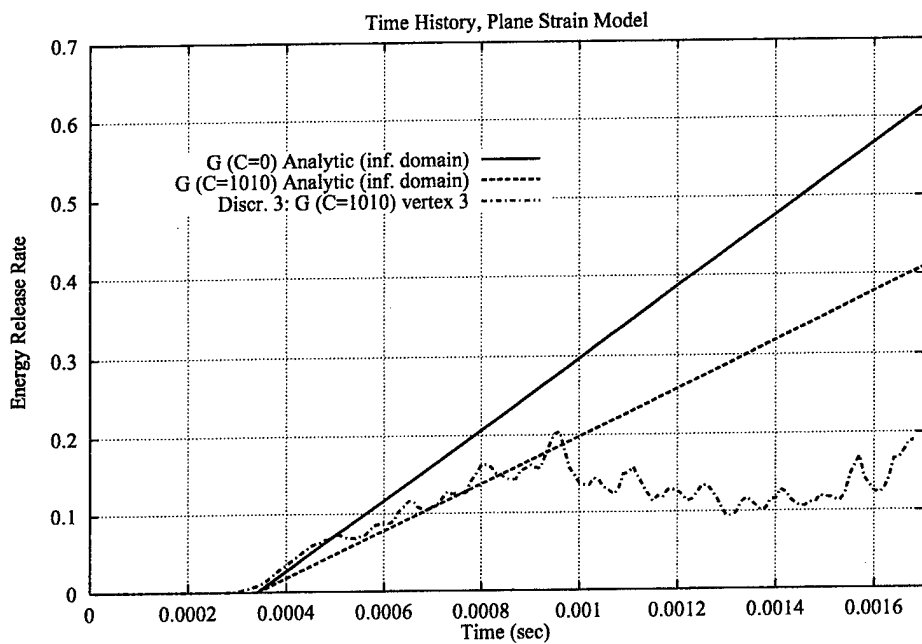


Figure 40: Time history for G using Discretization 3. Moving crack with advancement direction given by (10).

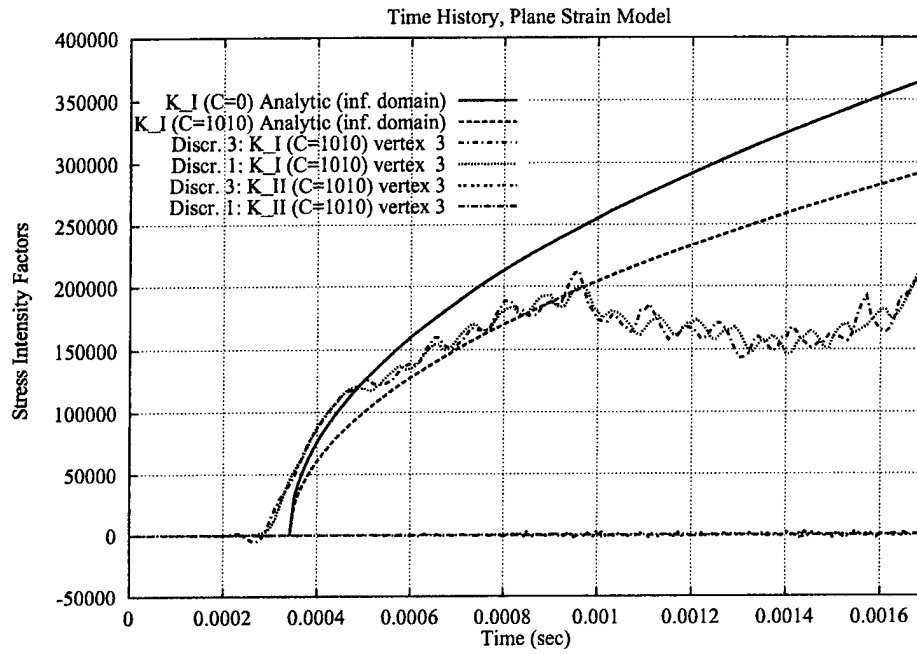


Figure 41: Comparison of results for K_I and K_{II} computed with Discretizations 1 and 3.

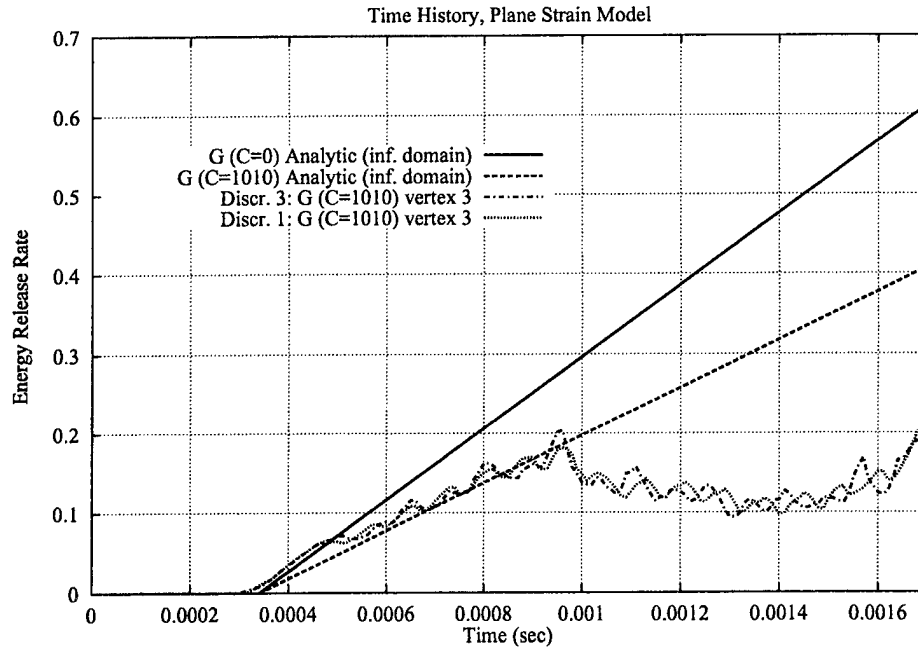


Figure 42: Comparison of results for G computed with Discretizations 1 and 3.

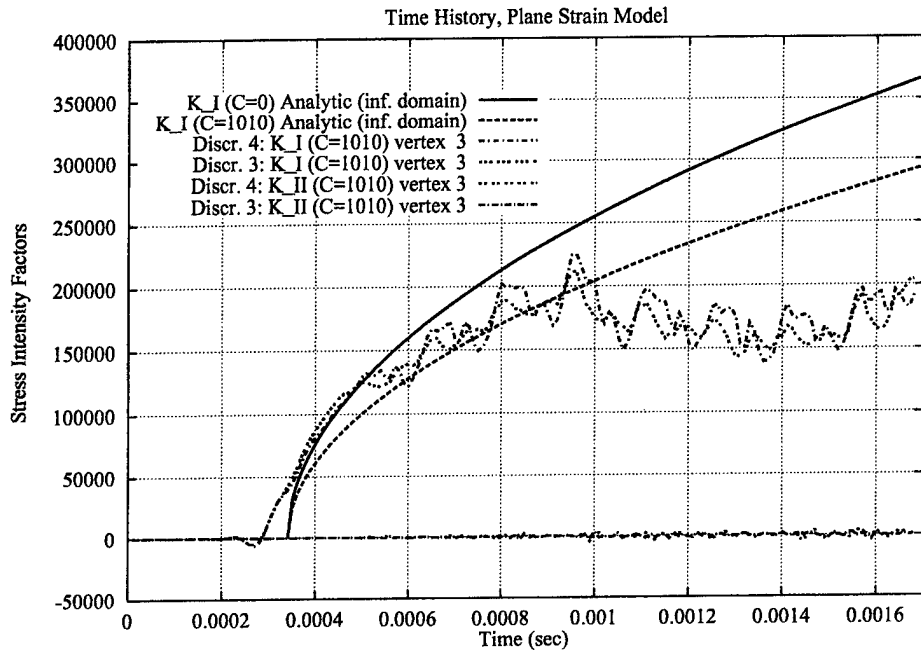


Figure 43: Time history for K_I and K_{II} using coarse mesh and $\mathbf{p}_c = \mathbf{0} + \mathbf{1}$, $\mathbf{p}_c = \mathbf{0} + (2, 2, 1)$ (Discretizations 3 and 4).

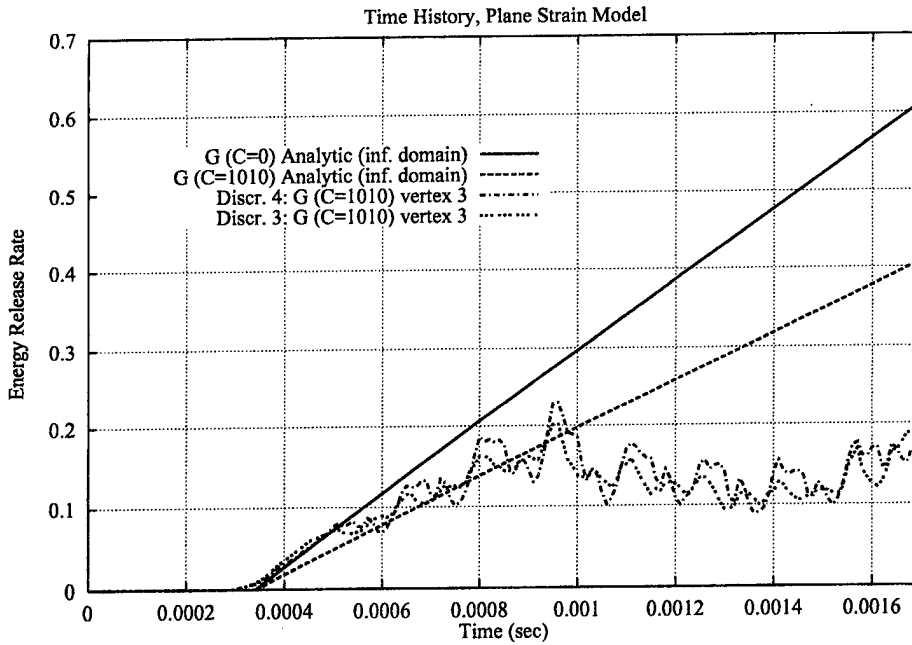


Figure 44: Time history for G using coarse mesh and $\mathbf{p}_c = \mathbf{0} + \mathbf{1}$, $\mathbf{p}_c = \mathbf{0} + (2, 2, 1)$ (Discretizations 3 and 4).

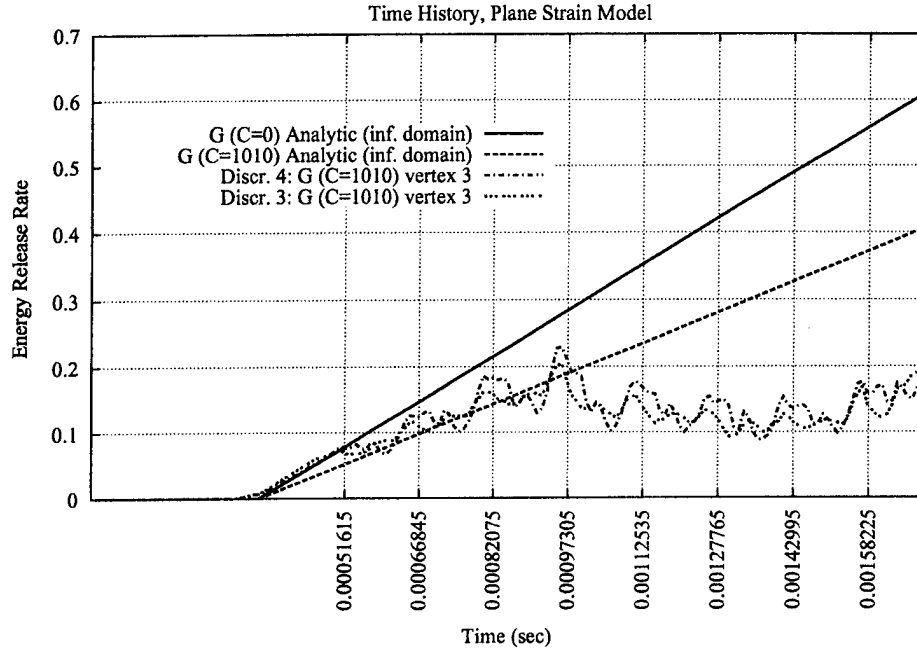


Figure 45: Time history for G using coarse mesh and $\mathbf{p}_c = \mathbf{0} + \mathbf{1}$, $\mathbf{p}_c = \mathbf{0} + (2, 2, 1)$ (Discretizations 3 and 4). The tic marks of the x-axis are placed exactly when the crack front crossed the boundary between two elements.

13.4 Edge Cracked Plate Under Asymmetric Impact

This example also shows the excellent agreement between solutions obtained with PHLEX-crack and analytical solutions. The problem is a plate containing an edge crack (Figure 46). Initially, the plate is stress-free and at rest. The asymmetric impact of a projectile on the cracked edge is simulated as a normal velocity imposed suddenly at $t=0$ on the side of the plate. The problem and its analytical solution were presented by Lee and Freund [26]. Under the current loading, the crack is subjected to negative mode I stress intensity factor and, therefore, will be stationary; the loading is predominantly mode II.

The problem was solved using both the finite element method and the EPM method. Figure 46 shows both models. Note that in the finite element model, successive levels of mesh refinement were needed at the crack tip in order to obtain accurate results. For the EPM model, singular functions were used at the crack tip. Model parameters and material properties were chosen as those used by Organ [44]:

- Young's modulus = $E = 200 \times 10^9 \text{ N/m}^2$
- Poisson's ratio = 0.25
- Mass density = 7833 Kg/m^3

- $\kappa \equiv$ Kosolov constant $= 3 - 4\nu = 2$ (for plane strain)
- Impact speed $= v_0 = 16.5$ m/sec
- Crack length $= a_0 = 1.0$ m
- Plate dimensions $= 4a_0 \times 6a_0$

Figure 47 presents the time histories of mode I and II stress intensity factors for all three solutions: Analytical, finite element, and EPM. The stress intensity factors were normalized by

$$\frac{-Ev_0\sqrt{a_0/\pi}}{2(1-\nu^2)c_d} = -0.17939 \times 10^9,$$

where

$$c_d = \sqrt{\frac{\kappa+1}{\kappa-1}} \sqrt{\frac{\mu}{\rho}} = 5535.3 \text{ m/s},$$

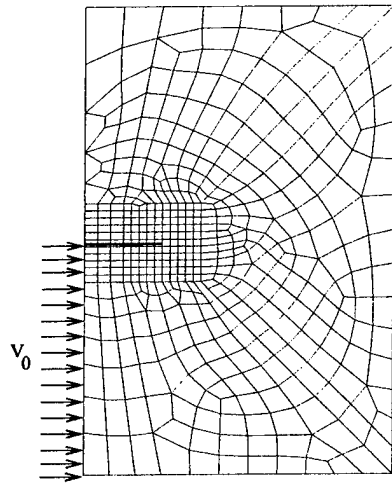
is the dilatational wave speed. Time was normalized with

$$a_0/c_d = 18.066 \times 10^{-5}$$

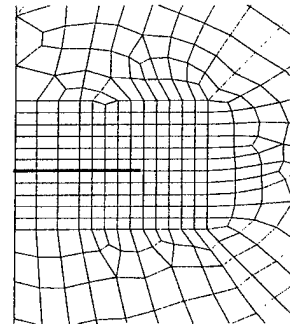
The two graphs of Figure 47 clearly show the excellent agreement among all three solutions. In this example, the natural and simplified inclusion of singular functions with the EPM approximation and the ability to place the crack anywhere in the model without modifying or refining it, are important advantages of EPM over FE method. The advantages of the EPM method, however, will become more evident in propagation problems where the crack can freely and arbitrarily advance inside the domain without the need for remeshing.

13.5 Welded T-Joint with a Crack

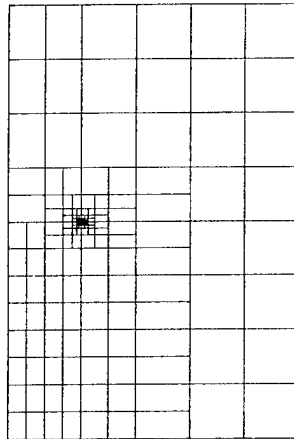
In this section we present a truly 3D example on using EPM for crack propagation. The example, as shown in Figure 48, is a beam with welded cross-section (T-section). An initial half-penny crack is placed longitudinally between the weld and the web as shown in the figure. The crack propagates due to an impact couple loading applied at time=0 at the end of the beam. Dynamic waves travel through the body. Once it reach the crack area, stresses increase sharply so that crack is propagated. It is assumed that both domain and loading are symmetric with respect to the other end of the beam and, therefore, only half of the domain is analyzed.



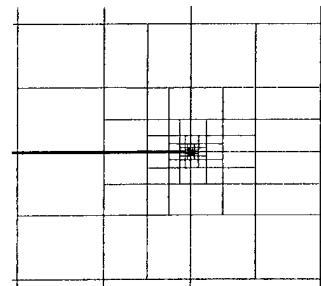
(a) EPM



(b) EPM in vicinity of crack tip



(c) FE



(d) FE in vicinity of crack tip

Figure 46: The grids used: (a,b) for EPM, and (c,d) for FE analyses.

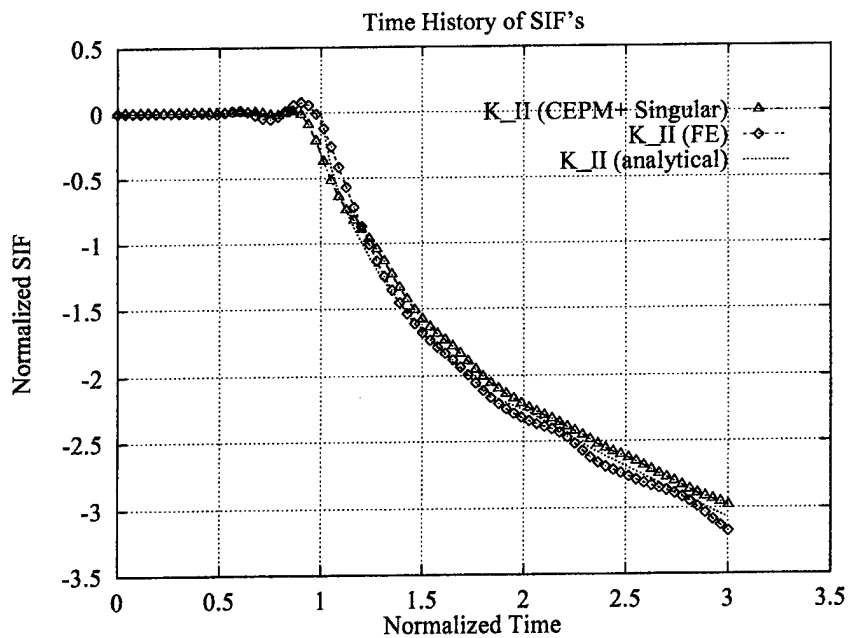
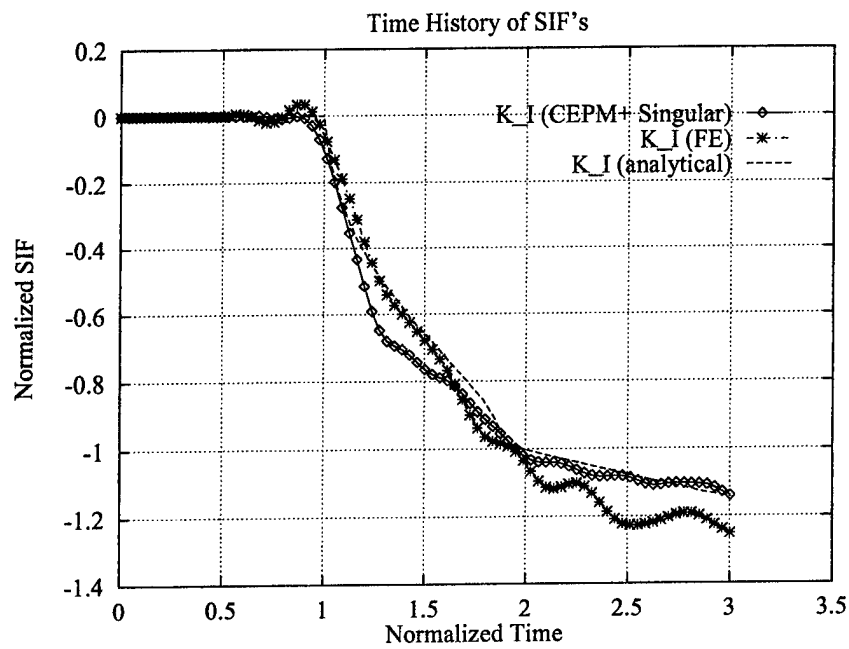


Figure 47: Comparisons of Stress Intensity Factors obtained from EPM, FE, and analytical solutions.

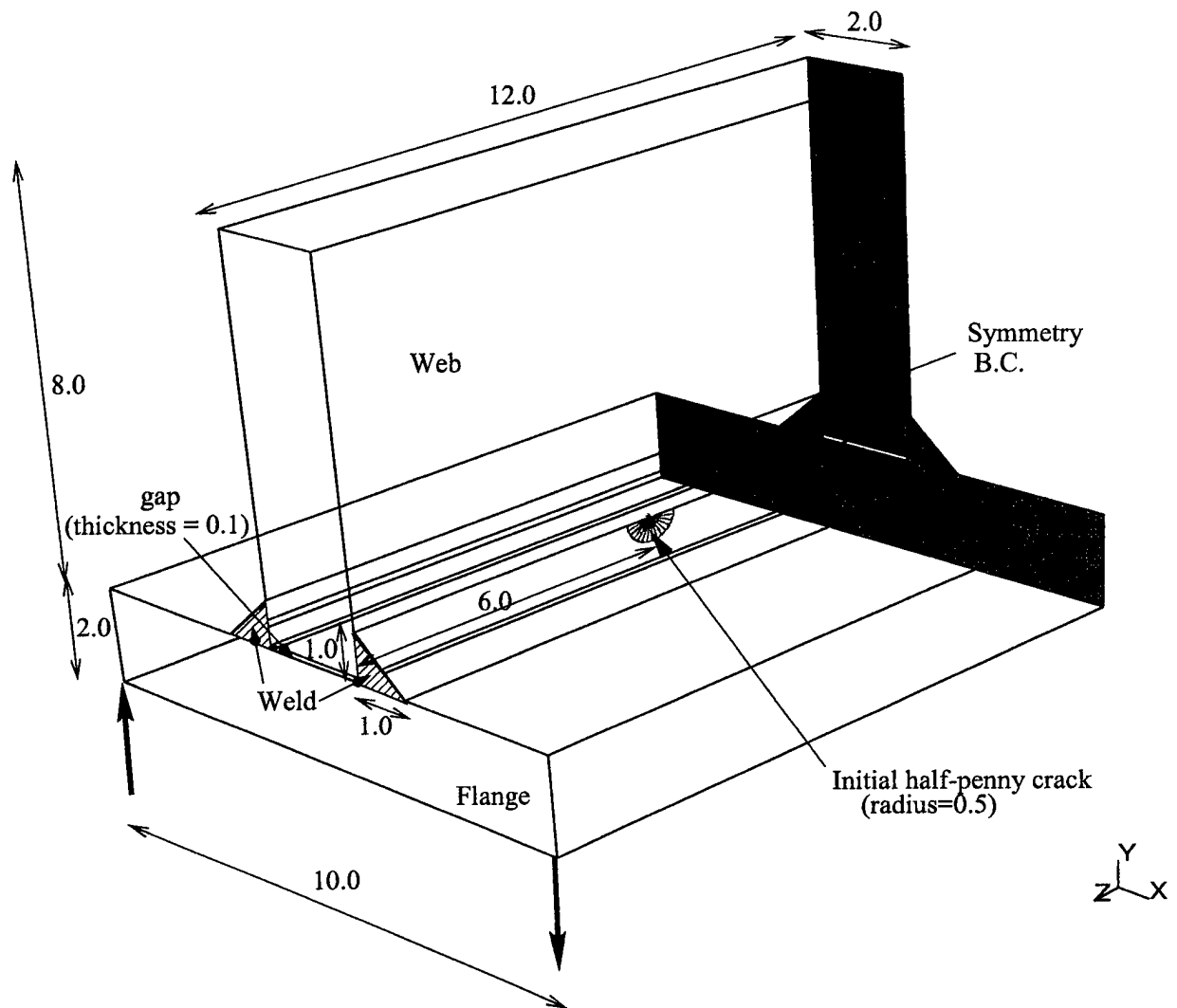


Figure 48: Welded T-section beam with a crack.

Both the web and the flange are made of the same linear elastic material with the following properties:

- Young's modulus = 200×10^9
- Poisson's ratio = 0.3
- Mass density = 7833

The weld material is assumed to be ten times stiffer than the latter (*i.e.*, Young's modulus = 200×10^{10}). The Freund propagation model is used to advance the crack (see section 6). The dynamic fracture properties and parameters used to propagate the crack are as follows:

- Least squares fit extraction domain: Cylinder of radius 0.5 and length 0.25
- Number of integration points in the r, θ and z directions: $\mathbf{n} = (5, 10, 3)$, respectively.
- Type of \mathbf{D} matrix is material matrix.
- Power for the weighting function = 3
- Dynamic fracture toughness in a pure mode I, $\hat{K}_{ID} = 75000$
- Crack speed limit = 1212

The couple is applied as two equal and opposite impact forces at the two corners of the edge cross section with a value of 5×10^6 each.

The initial crack surface definition is shown in Figure 48. Note that the web is not in contact with the flange, but rather a small gap exists between the two. The domain is discretized for EPM as shown in Figure 49. The figure shows the integration cells; the meshless nodes are located at the corners of the these cells. Note that the grid is made finer around the crack area. Linear approximation is used over all the domain.

A transient dynamic analysis is performed on the model described above. Newmark method is used to march the solution over time. Figures 50, 51, and 52 represent the crack surface at times 0.0015, 0.0020, and 0.0030 respectively.

13.6 Welded T-Joint with a Fully Embedded Crack

This example shows identical domain as in the previous example, but the initial crack has been changed to full disc and moved to the inside of the domain. Fig. 53 shows the position

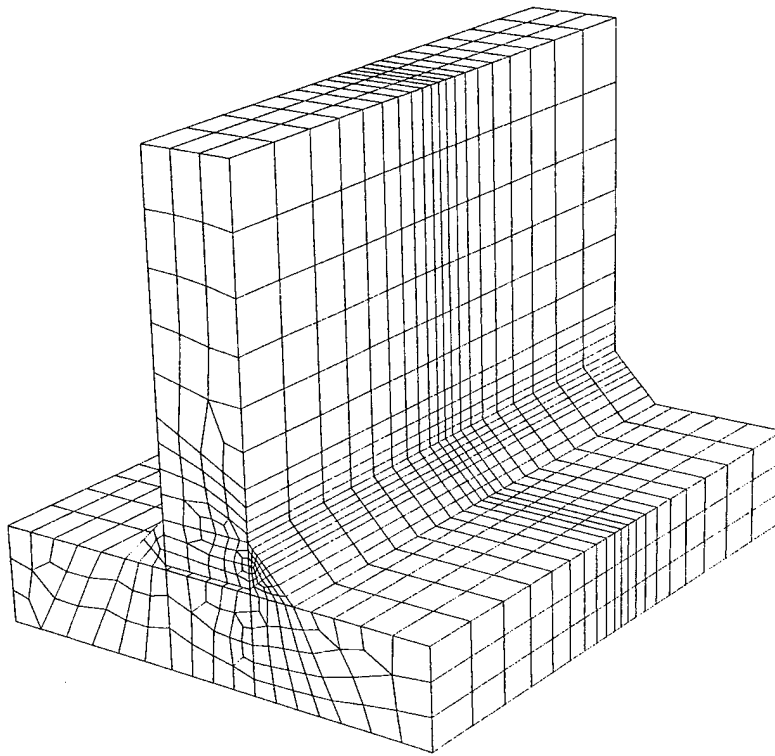


Figure 49: EPM discretization for welded T-section beam. $\begin{matrix} y \\ | \\ z-x \end{matrix}$

of the crack, and Fig. 54 shows the crack at the time 0.0021. This crack has been formed due to 5 consecutive shock waves, between them the speed of crack front slowed down and even stopped (this is visible as darker bands of edges inside the crack surface). History of crack propagation is shown in Fig. 55.

This example shows also start of numerical instability in the crack. Details of this instability, and practical means to control it have been discussed in Section. 12.

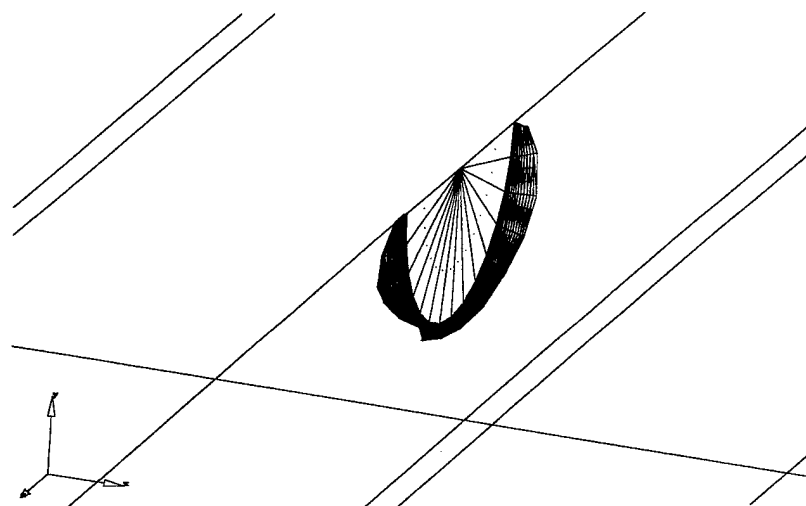


Figure 50: Crack surface of welded T-section example at time 0.0015.

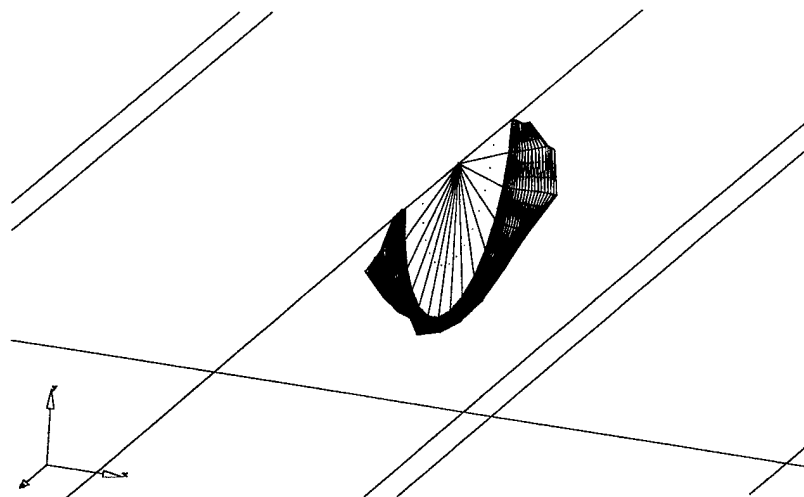


Figure 51: Crack surface of welded T-section example at time 0.0020.

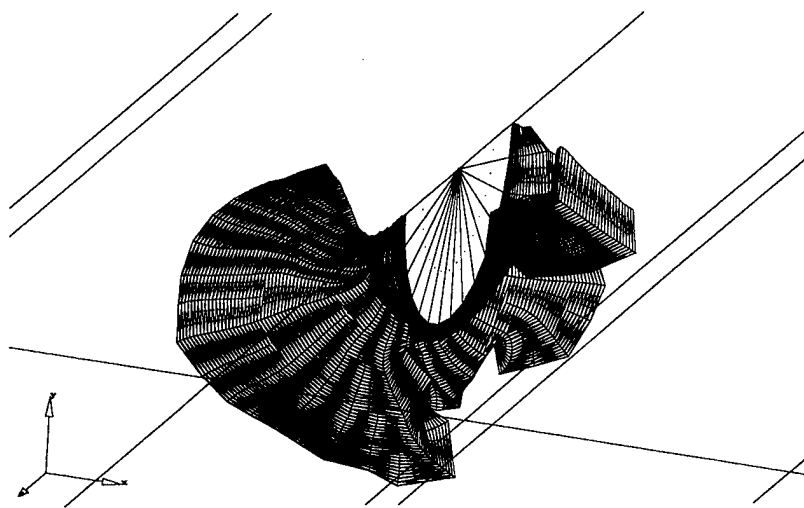
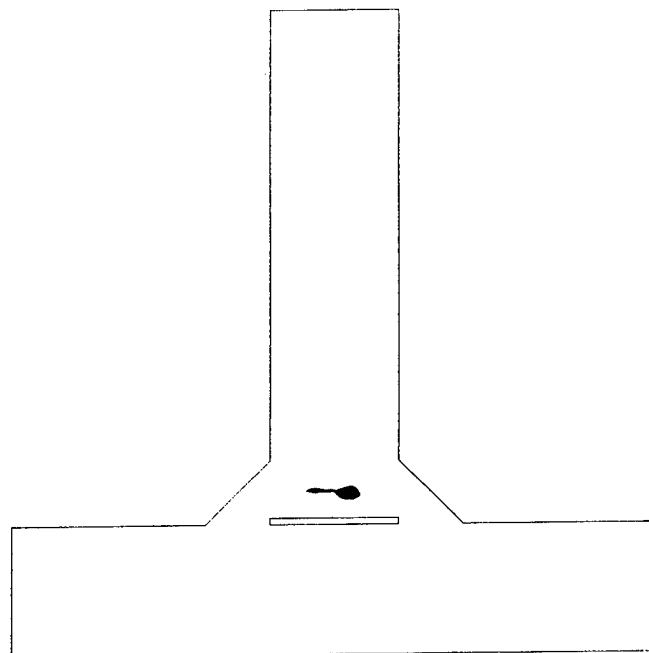
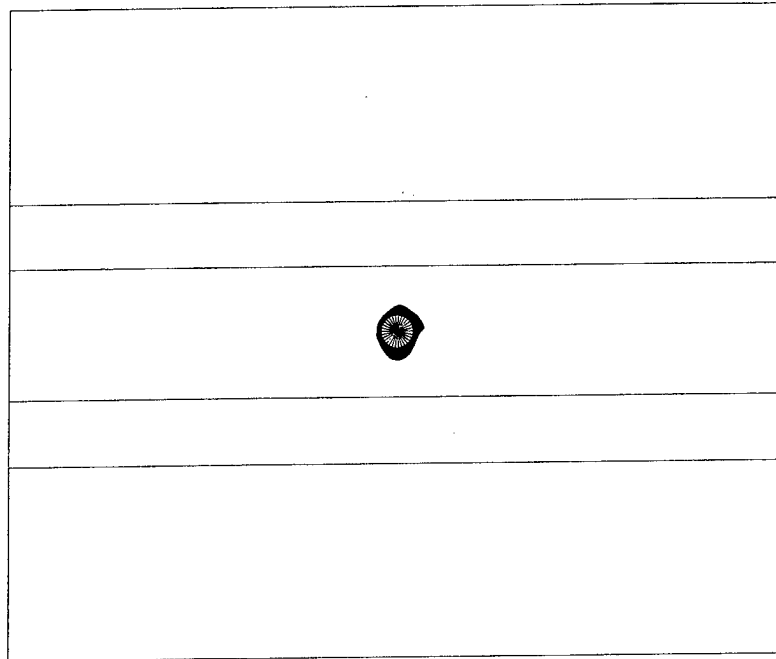


Figure 52: Crack surface of welded T-section example at time 0.0030.



X
Y Z

Figure 53: Welded T-section beam with a penny-shaped crack.

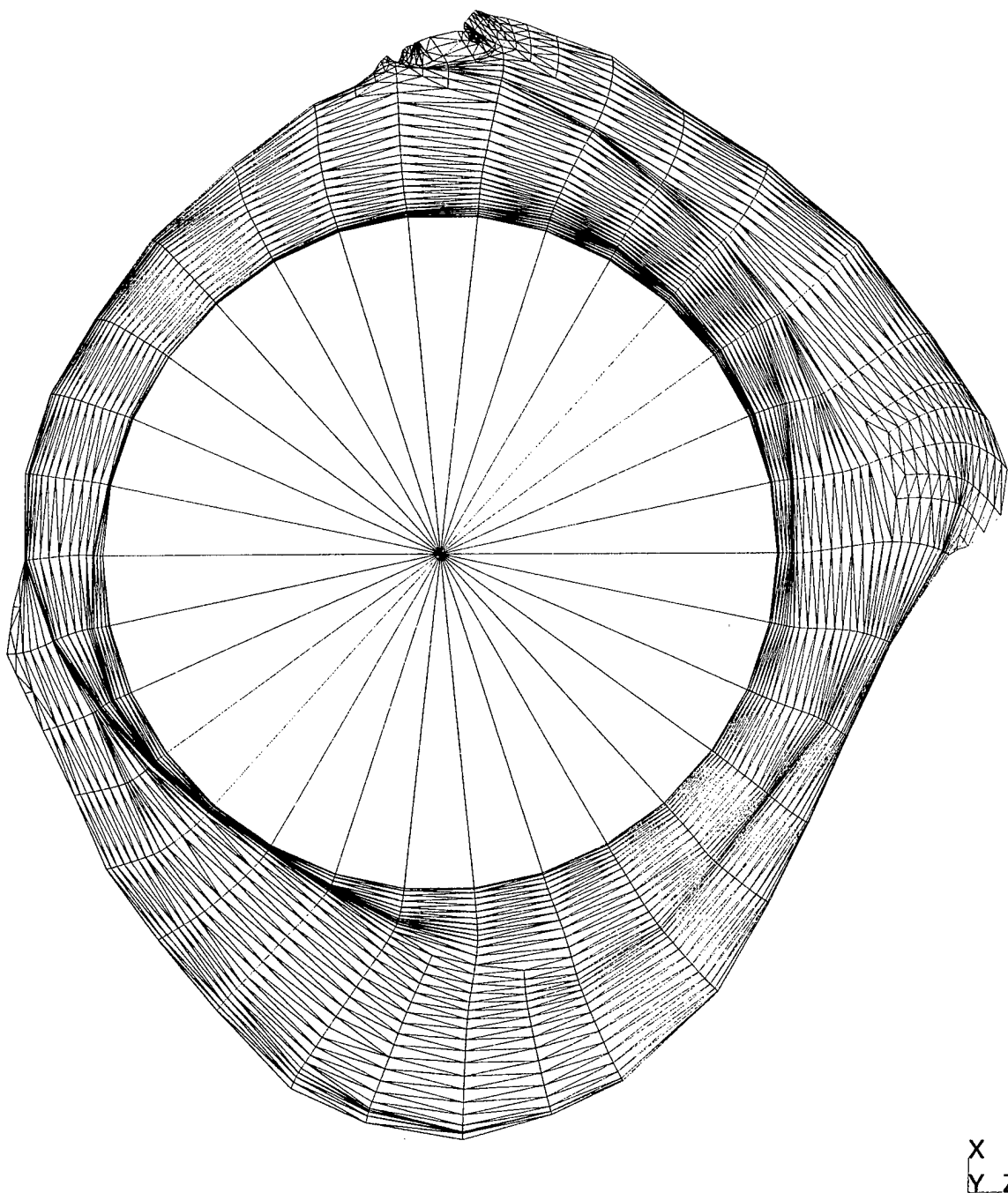


Figure 54: Top view of the penny shape crack in a welded T-section beam. (Shock waves are coming along Z-axis)

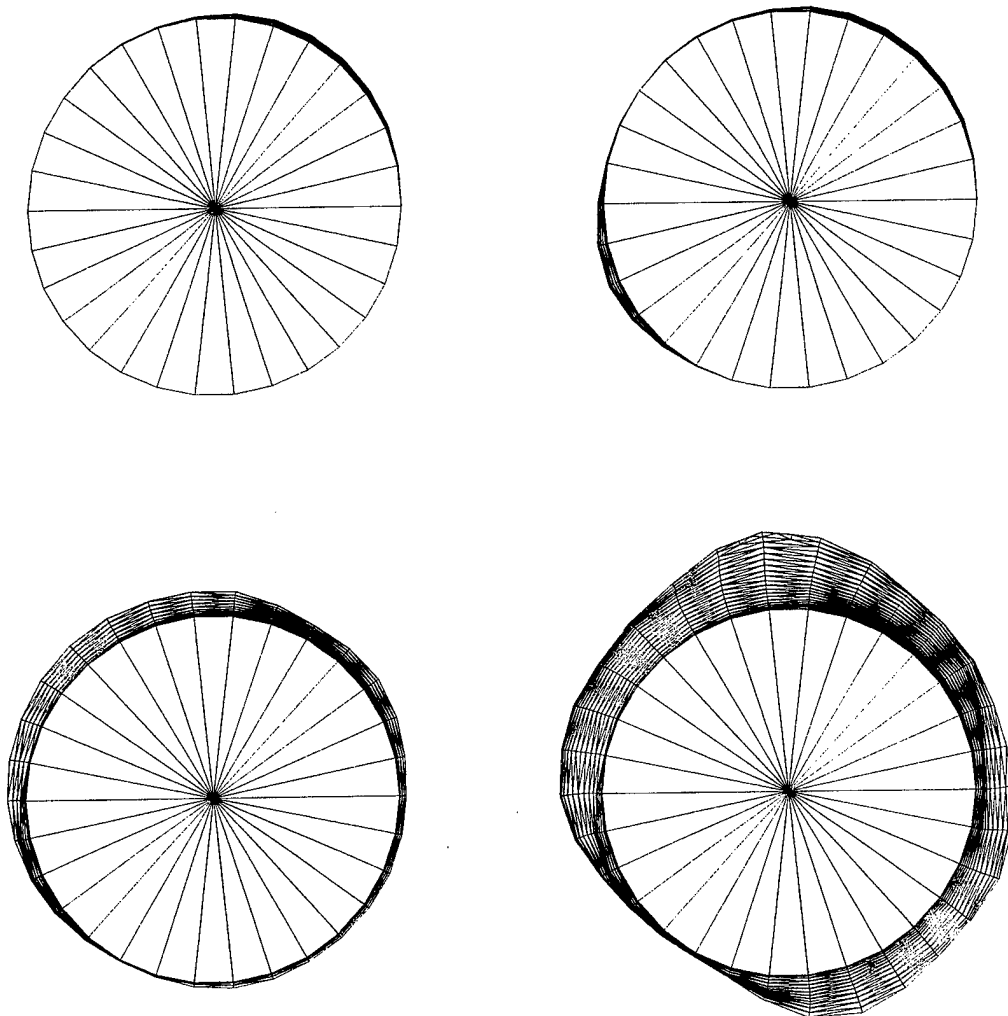


Figure 55: Penny shape crack in a welded T-section beam at times 0.0014, 0.0016, 0.0018, 0.0019

13.7 Section of Infinite Cylinder

This example shows the capability of the code to perform analysis using classical finite elements and EPM ICell mesh. The initial model consists of half of the cylinder, with a rib modeled using shell elements (QUAD4). Appropriate symmetry boundary conditions are applied to the shell and EPM elements. The domain and boundary conditions are shown in Fig. 56.

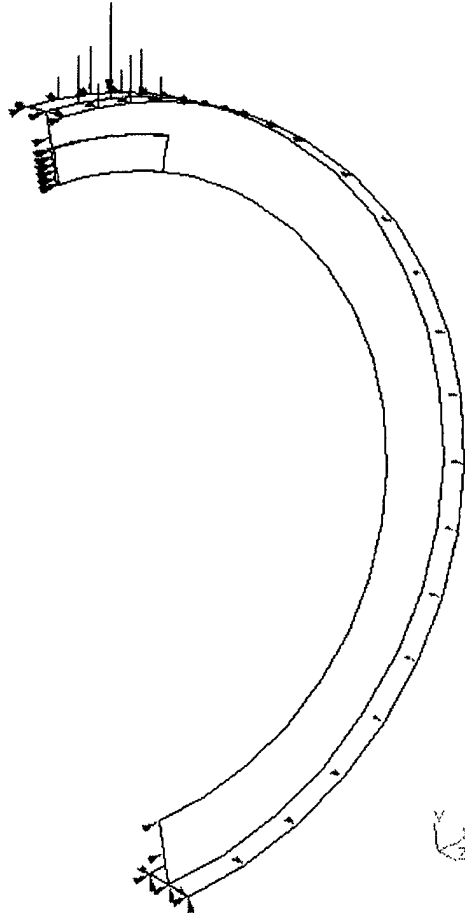


Figure 56: Domain and boundary conditions for the infinite cylinder problem.

The physical and material parameters for this problem are meant to simulate a steel shell but are non-dimensionalized.

- Geometric properties:
 - Shell radius = 1.0
 - Shell and rib thickness = 0.01
 - Axial width = 0.175

- Rib inner radius = 0.825
- Material properties:
 - Young's modulus $E = 98.125$
 - Poisson's ratio $\nu = 0.3$
 - Mass density $\rho = 7.85$
 - Dynamic fracture toughness $K_{ID} = 1.455 \times 10^{-5}$
 - Rayleigh wave speed $C_R = 2.0335$

The load is applied at time $t = 0$ as shown in Fig. 56. The point loads correspond to a uniform pressure of magnitude 10^{-2} applied to the faces of the elements shown in Fig. 56.

The following sequence of problems has been solved for this model:

- 1 Coarse shell mesh (18 x 2 x 2 linear elements, i.e. 18 elements along the arc, 2 in radial and 2 in the Z direction)
- 2 Fine shell mesh (54 x 6 x 6 linear elements)
- 3 Mixed mesh with part of the rib modeled using EPM approximation (Figs. 57 and 58). The shell elements are linear while the ICells are quadratic in the X and Y directions and linear in the Z direction.
- 4 Mixed mesh with crack (Fig. 59). The same p -order as in the previous problem is used here.

The dilatational wave speed is given by

$$C_d = \sqrt{\frac{\kappa + 1}{\kappa - 1} \frac{\mu}{\rho}} = 4.102$$

where $\kappa = 3 - 4\nu$ and $\mu = \frac{E}{2(1-\nu)}$.

In problems 3 and 4, the time step was chosen equal to $\Delta t = 6 \times 10^4$ which is about half of h_{min}/C_d where h_{min} is the smallest element size in the mesh.

In the case of problem 4 the crack speed is prescribed as

$$C(t) = H(t - 0.06) \frac{C_R}{6} = H(t - 0.06) 0.3389$$

By comparing results of shell-only models it has been determined that the model is sufficient to obtain qualitative results, but further refinement (i.e. finer model) will probably

result in better values of the stresses and crack behavior. The results will not be presented here in details, only following figures are enclosed to illustrate the results. Fig. 60 shows stresses at the time 0.00009 in the model without the crack, Figs. 61 and 62 show displacements and stresses in the presence of stationary crack at the time 0.0001. Figures 63, 64 and 65 show the crack growth at the times $t = 0.072$, 0.102 , 0.138 , respectively.

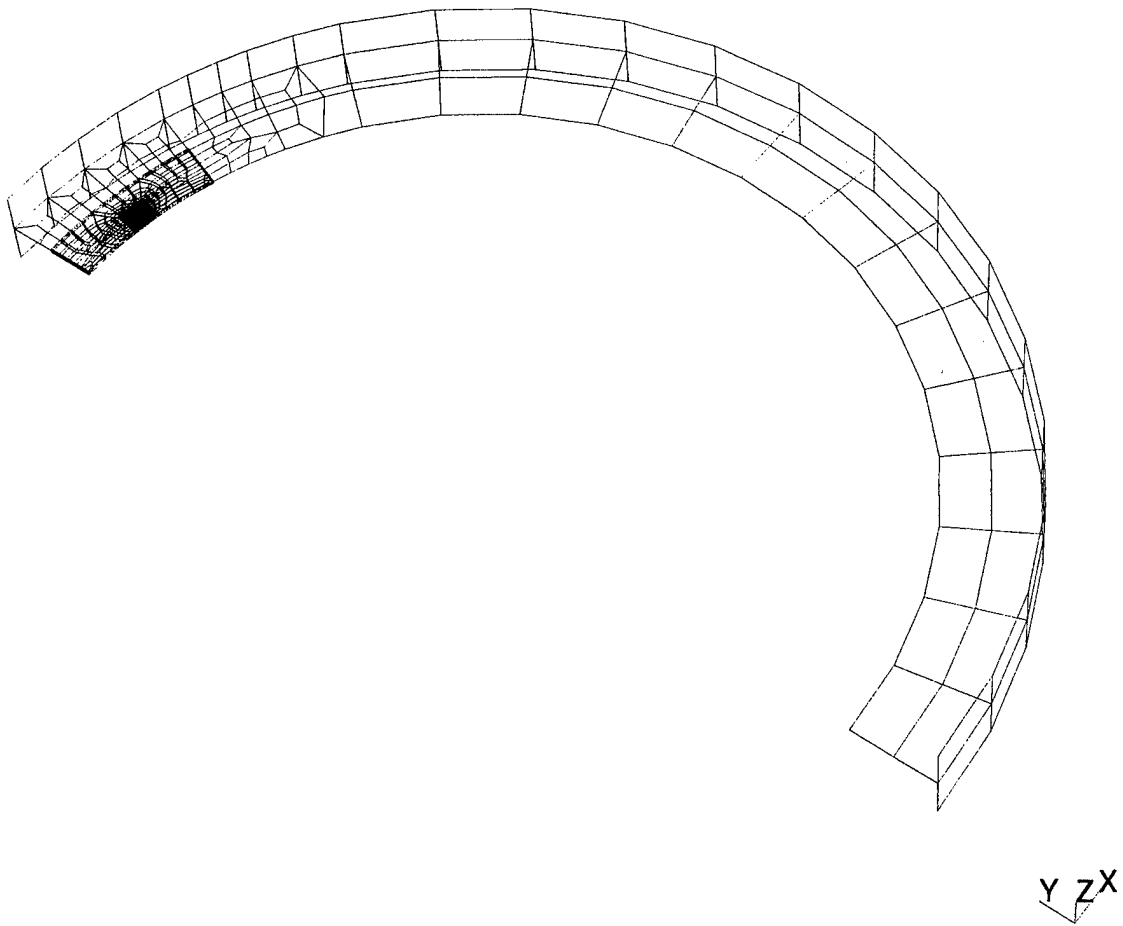


Figure 57: Infinite cylinder model using classical finite elements and EPM ICells.

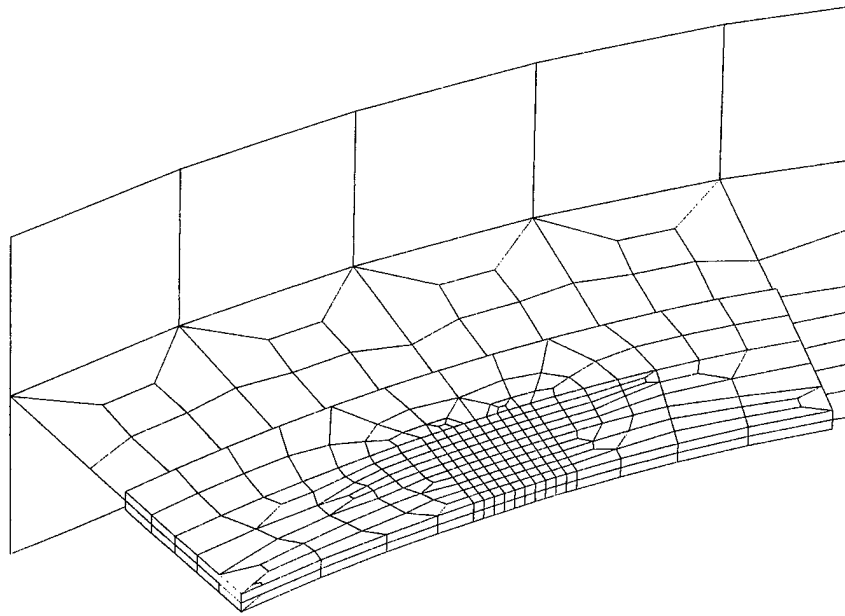


Figure 58: EPM subsection of mixed infinite cylinder model.

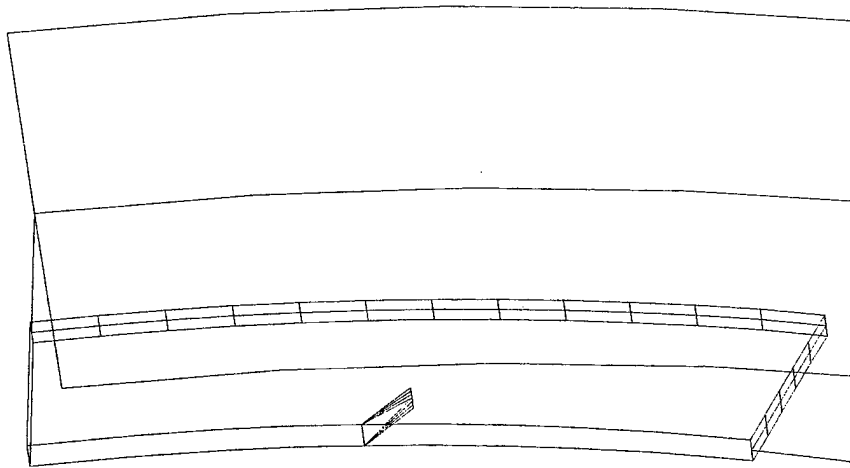


Figure 59: Infinite cylinder with the initial crack.

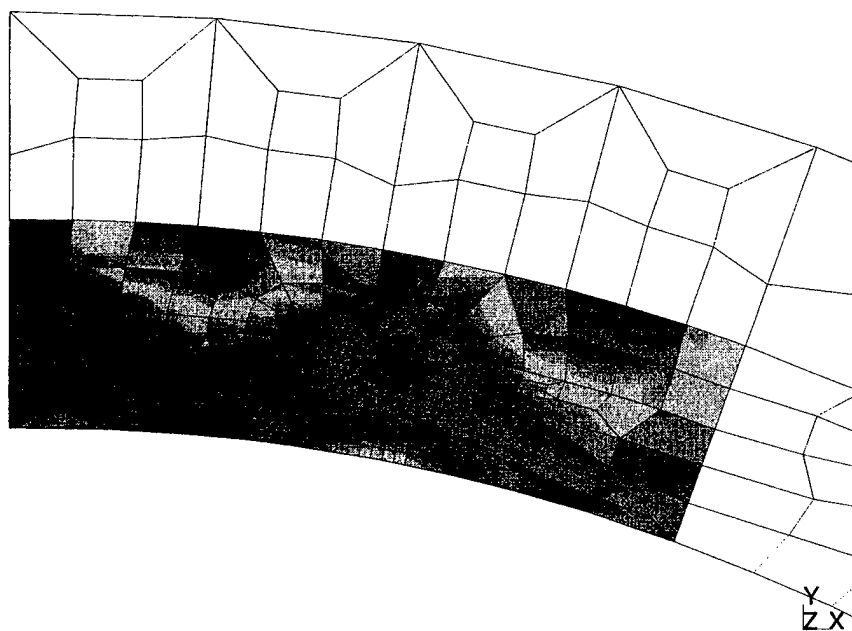


Figure 60: Infinite cylinder: Stresses in the model without the crack, time = $9e-5$

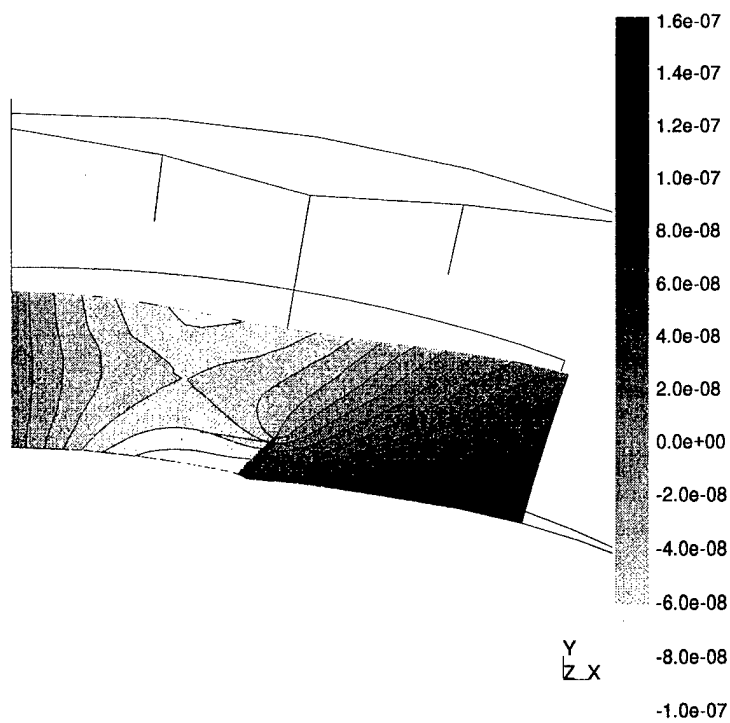


Figure 61: Infinite cylinder: X Displacements in the model with the crack, time = 0.0001

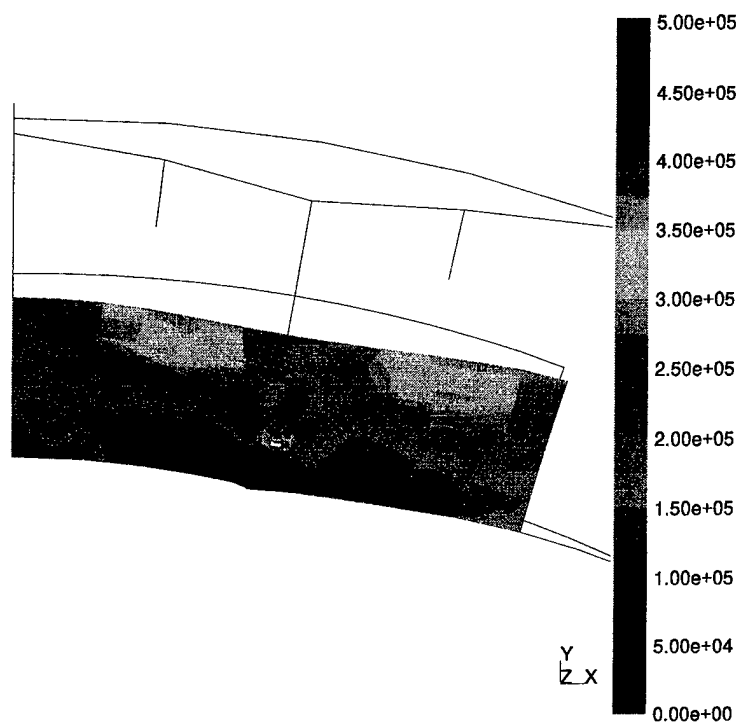


Figure 62: Infinite cylinder: Stresses in the model with the crack, time = 0.0001

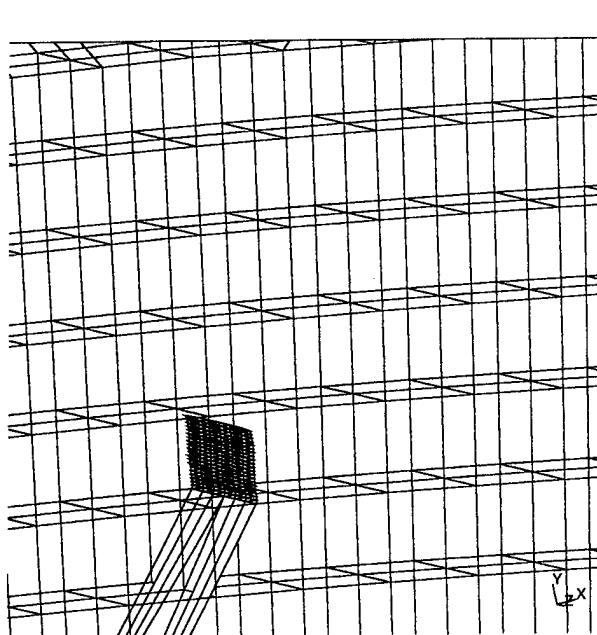


Figure 63: Infinite cylinder: Crack growth at the time $t = 0.072$.

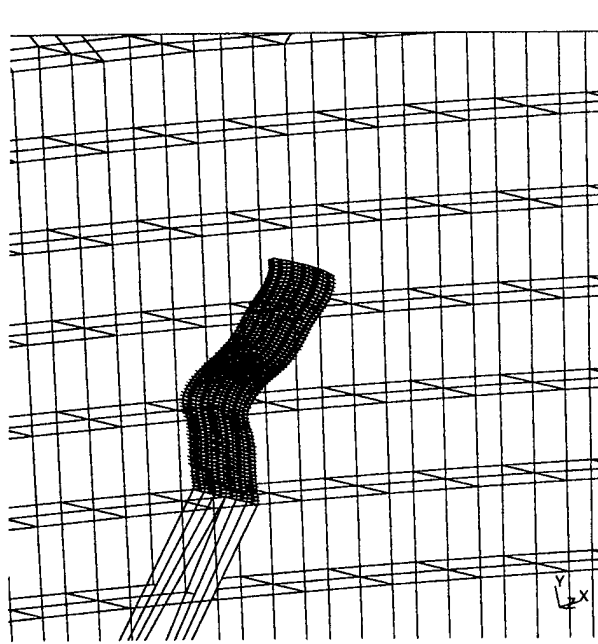


Figure 64: Infinite cylinder: Crack growth at the time $t = 0.102$.

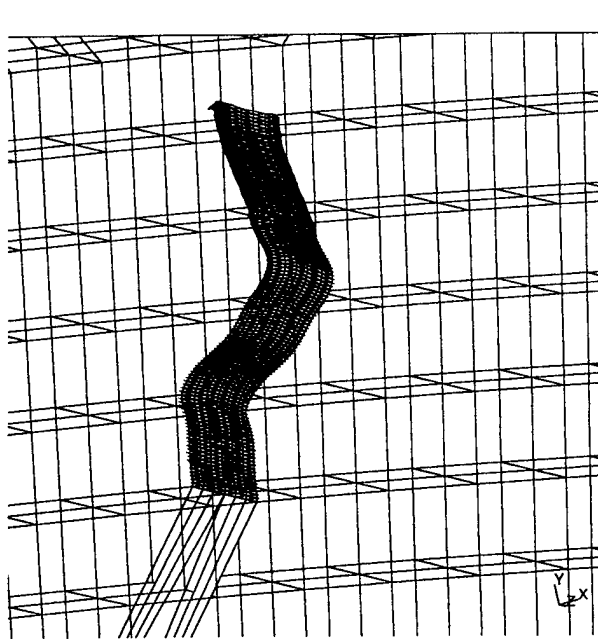


Figure 65: Infinite cylinder: Crack growth at the time $t = 0.138$.

References

- [1] J. Alonso, L. Martinelli, and A. Jameson. Multigrid unsteady navier-stokes calculations with aeroelastic applications. *AIAA Paper*, 95-0048:1-15, 1995.
- [2] I. Babuska, G. Caloz, and Osborn J. E. Special finite element methods for a class of second order elliptic problems with rough coefficients. *SIAM J. Numerical Analysis*, 31(4):745-981, 1994.
- [3] I. Babuška and J. M. Melenk. The partition of unity finite element method. *International Journal for Numerical Methods in Engineering*, 40:727-758, 1997.
- [4] T. Belytschko, Y. Y. Lu, and L. Gu. Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 37:229-256, 1994.
- [5] T. Belytschko and M. Tabbara. Dynamic fracture using element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 39:923-938, 1996.
- [6] Bendiksen. A new approach to computational aeroelasticity. *AIAA Publication*, 91-0931-CP:1712-1727, 1991.
- [7] R.L. Bisplinghoff, H. Ashley, and R.L. Halfman. *Aeroelasticity*. Addison-Wesley, Cambridge, MA, 1955.
- [8] H. M. Blackburn and G. E. Karniadakis. Two- and three- dimensional vortex-induced vibration of a circular cylinder. 1993. ISOPE-93 Conference, Singapore, 1993.
- [9] L. Demkowicz, A. Karafiat, and T. Liszka. On some convergence results for FDM with irregular mesh. *Computer Methods in Applied Mechanics and Engineering*, 42:343-355, 1984.
- [10] Jr. DeRuntz, John A. The underwater shock analysis code and its applications. In *Shock and Vibration Symposium Proceedings*, volume I, pages 89-107. David Taylor Research Center, 1989.
- [11] E.H. Dowell. Flutter of a buckled plate as an example of chaotic motion of a deterministic autonomous system. *Journ. of Sound and Vibration*, 147:1-38, 1982.
- [12] E.H. Dowell, H.C. Curtis, R.H. Scanlan, and F. Sisto. *A modern course in aeroelasticity*. Sijthoff & Noordhoff, Netherlands, 1980.
- [13] C. A. Duarte, O. N. Hamzeh, T. J. Liszka, and W. W. Tworzydło. The element partition method for the simulation of three-dimensional crack propagation. Submitted, July 1999.
- [14] C. A. Duarte, O. N. Hamzeh, T. J. Liszka, and W. W. Tworzydło. Simulation of three-dimensional crack growth under impact loads using a partition of unity method. Submitted, July 1999.

- [15] C. A. M. Duarte and J. T. Oden. Hp clouds—a meshless method to solve boundary-value problems. Technical Report 95-05, TICAM, The University of Texas at Austin, May 1995.
- [16] C. A. M. Duarte and J. T. Oden. An *hp* adaptive method using clouds. *Computer Methods in Applied Mechanics and Engineering*, 139:237–262, 1996.
- [17] C. A. M. Duarte and J. T. Oden. *Hp* clouds—an *hp* meshless method. *Numerical Methods for Partial Differential Equations*, 12:673–705, 1996.
- [18] J. Dugundji. Nonlinear problems of aeroelasticity. In S.A. Alturi, editor, *Computational Nonlinear Mechanics in Aerospace Engineering*, volume 146, pages 127–155. AIAA, 1992. Progress in Astronautics and Aeronautics.
- [19] T. J. Liszka *et al.* Prophlex – an *hp*-adaptive finite element kernel for solving coupled systems of partial differential equations in computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 139:151–271, 1997.
- [20] L. B. Freund. *Dynamic Fracture Mechanics*. Cambridge University Press, Cambridge, 1990.
- [21] R. Gallego and J. Dominguez. Dynamic crack propagation analysis by moving singular boundary elements. *J. Appl. Mech, Trans. ASME*, 59:S158–S162, 1992.
- [22] M.E. Gurtin. *An Introduction to Continuum Mechanics*. Academic Press, 1981.
- [23] C. Y. Huang, W. W. Tworzydło, S. R. Kennon, and J. T. Oden. Progress in *h*- and *hp*-adaptive methods of flow analysis in propulsion systems. Technical report, 9th NASA CFD Workshop, Marshall Space Flight Center, April 1991.
- [24] P. Krysl and T. Belytschko. A three-dimensional crack growing under a tensile pulse: Simulation by the element free Galerkin method. *Int. J. Solids and Structures*, to appear.
- [25] P. Lancaster and K. Salkauskas. Surfaces generated by moving least squares methods. *Mathematics of Computation*, 37(155):141–158, 1981.
- [26] Y.J. Lee and L. B. Freund. Fracture initiation due to asymmetric impact loading of an edge cracked plate. *Transactions of the ASME*, 57:104–111, 1990.
- [27] T. Liszka and W. W. Tworzydło. An *hp* adaptive cloud method for a class of second order pde's. In *3rd U.S. National Congress on Comp. Mechanics*, Dallas, USA, 1995.
- [28] Tadeusz Liszka. An interpolation method for an irregular net of nodes. *International Journal for Numerical Methods in Engineering*, 20:1599–1612, 1984.
- [29] Tadeusz Liszka and J. Orkisz. The finite difference method at arbitrary irregular grids and its application in applied mechanics. *Computers and Structures*, 11:83–95, 1980.

- [30] T.J. Liszka, C.A.M. Duarte, and W.W. Tworzydło. h - p -meshless cloud method. *Computer Methods in Applied Mechanics and Engineering*, 139:263–288, 1996.
- [31] W. K. Liu, S. Jun, and Y. F. Zhang. Reproducing kernel particle methods. *International Journal for Numerical Methods in Engineering*, 20:1081–1106, 1995.
- [32] Y. Y. Lu, T. Belytschko, and M. Tabbara. Element-free Galerkin method for wave propagation and dynamic fracture. *Computer Methods in Applied Mechanics and Engineering*, 126:131–153, 1995.
- [33] J. M. Melenk and I. Babuška. The partition of unity finite element method: Basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, 139:289–314, 1996.
- [34] S. A. Morton, R. B. Melville, and Visbal M. R. Accuracy and coupling issues of aeroelastic navier-stokes solutions on deforming meshes. *AIAA Paper*, 97-1085:1–11, 1997.
- [35] B. Nayroles, G. Touzot, and P. Villon. Generalizing the finite element method: Diffuse approximation and diffuse elements. *Computational Mechanics*, 10:307–318, 1992.
- [36] J. T. Oden and J. M. Bass. New developments in adaptive methods for computational fluid dynamics. In *Proceedings: Ninth International Conference on Computing Methods in Applied Sciences and Engineering, Paris, France*. SIAM Journal, 1990.
- [37] J. T. Oden and C. A. Duarte. Clouds, cracks and fem's. In B. D. Reddy, editor, *Recent Developments in Computational and Applied Mechanics*, pages 302–321, Barcelona, Spain, 1997. International Center for Numerical Methods in Engineering, CIMNE.
- [38] J. T. Oden, C. A. Duarte, and O. C. Zienkiewicz. A new cloud-based hp finite element method. *Computer Methods in Applied Mechanics and Engineering*, 153:117–126, 1998.
- [39] J. T. Oden and C. A. M. Duarte. Solution of singular problems using hp clouds. In J. R. Whiteman, editor, *The Mathematics of Finite Elements and Applications 96*, pages 35–54, New York, NY, 1997. John Wiley & Sons.
- [40] J. T. Oden, T. Liszka, and W. Wu. An hp adaptive finite element method for incompressible viscous flows. In *The Mathematics of Finite Elements and Applications*, pages 13–54. Academic Press, 1991.
- [41] J. T. Oden, T. Strouboulis, and J. M. Bass. Paradigmatic error calculation for adaptive finite element approximations of convection dominated flows. In T. Tezduyar and T. J. R. Hughes, editors, *Recent Advances in Computational Fluid Dynamics*, volume 95, pages 147–162. ASME Publications, N.Y., 1989.
- [42] J.T. Oden, L. Demkowicz, T. Liszka, and W. Rachowicz. Adaptive finite element methods for compressible and incompressible flows. *Computing Systems in Engineering*, 1:523–534, 1990.

- [43] J.T. Oden, S.R. Kennon, W.W. Tworzydło, J.M. Bass, and C.W. Berry. Progress on adaptive hp-finite element methods for the incompressible navier-stokes equations. *Computational Mechanics*, pages 421–432, 1993.
- [44] D. J. Organ. *Numerical Solutions to Dynamic Fracture Problems Using the Element-Free Galerkin Method*. PhD dissertation, Northwestern University, 1996.
- [45] D. Shepard. A two-dimensional function for irregularly spaced data. In *ACM National Conference*, pages 517–524, 1968.
- [46] B. A. Szabo and I. Babuska. Computation of the amplitude of stress singular terms for cracks and reentrant corners. In T. A. Cruse, editor, *Fracture Mechanics: Nineteenth Symposium, ASTM STP 969*, pages 101–124, 1988.
- [47] Barna Szabo and Ivo Babuska. *Finite Element Analysis*. John Wiley and Sons, New York, 1991.
- [48] H. Tada, P. Paris, and G. Irwin. *The Stress Analysis of Cracks Handbook*. Del Research Corporation, St. Louis, Missouri, 1973.
- [49] E. A. Thornton, J. T. Oden, W. W. Tworzydło, and S. K. Youn. Thermo-viscoplastic analysis of hypersonic structures subjected to severe aerodynamic heating. *Journal of Aircraft*, 27:826–835, 1990.
- [50] T.L.Geers. Residual potential and approximate methods for three-dimensional fluid interaction problems. *J. Acoust. Soc. Am.*, 49:1505–1510, 1971.
- [51] T.L.Geers. Doubly asymptotic approximations for transient motions of submerged structures. *J. Acoust. Soc. Am.*, 64:1500–1508, 1978.
- [52] W. Tworzydło. Analysis of large deformations of membrane shells by the generalized finite difference method. *Computers and Structures*, 27(1):39–59, 1987.
- [53] W. W. Tworzydło, C. Y. Huang, and J. T. Oden. Adaptive implicit / explicit finite element methods for axisymmetric viscous turbulent flows with moving boundaries. *Computer Methods in Applied Mechanics and Engineering*, 97:245–288, 1992.
- [54] W. W. Tworzydło, J. T. Oden, and E. A. Thornton. Adaptive implicit-explicit finite element method for compressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 95:397–440, 1992.
- [55] W.W. Tworzydło, J.T. Oden, Chang H.J., C.S. Venkat, C. Baumann, and M.R. Reddy. Dynamic instabilities in flow-structure interactions, flutter and buffeting. Technical Report AFOSR Contract F49620-95-C-0007, Computational Mechanics Company, 1998. final report to Air Force Office of Scientific Research.

- [56] L.N. Virgin and E.H. Dowell. Nonlinear aeroelasticity and chaos. In S.A. Alturi, editor, *Computational Nonlinear Mechanics in Aerospace Engineering*, volume 146, pages 531–545. AIAA, 1992. Progress in Astronautics and Aeronautics.